ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA

DIPARTIMENTO di

INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE

"Guglielmo Marconi"

DEI

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'ENERGIA ELETTRICA

TESI DI LAUREA

in

Metodologie di progettazione delle macchine elettriche

Sviluppo di Schemi di Controllo per l'Emulazione di Macchine Elettriche in Sistemi di Power Hardware in the Loop

CANDIDATO

NICOLO' ANGELI

RELATORE

Dott. Ing. GABRIELE RIZZOLI

CORRELATORI

Dott. Ing. MATTEO RAMPONI

Dott. Ing. LUCA VANCINI

Anno Accademico 2024/2025

Sessione III

Abstract

Questa tesi di laurea ha come obiettivo principale quello di confrontare e valutare schemi di controllo per l'emulazione di macchine elettriche lineari e non lineari, attraverso software di modellistica e simulazione per gli azionamenti elettrici: *Simulink* (MathWorks) e *PLECS* (Plexim).

Nella prima parte del lavoro vengono implementati e analizzati differenti modelli di motori elettrici tramite il software PLECS, con l'obiettivo di studiarne il comportamento dinamico, le caratteristiche elettromeccaniche e le principali strategie di controllo applicabili.

La seconda parte della tesi è dedicata alla validazione sperimentale dei modelli sviluppati mediante l'impiego di un'architettura Power Hardware-in-the-Loop (PHIL). Tale approccio permette di inserire componenti hardware reali all'interno di un ambiente di simulazione in tempo reale, consentendo il confronto diretto tra le prestazioni dei modelli simulati e il comportamento effettivo del sistema. L'obiettivo è verificare l'accuratezza della modellazione e valutare le potenzialità dell'ambiente PHIL come strumento di supporto alla progettazione e alla sperimentazione di sistemi di azionamento basati su motori lineari e non lineari.

Sommario

Indice delle figure

Capitolo 1

Introduzione

Negli ultimi anni, la crescente diffusione di sistemi elettrici avanzati, quali convertitori di potenza, macchine elettriche e reti intelligenti, ha reso sempre più necessario disporre di strumenti di sviluppo e validazione che garantiscano affidabilità, sicurezza e rapidità nella fase di progettazione. In questo contesto, le tecniche di Hardware-in-the-Loop (HIL) e, in particolare, la loro estensione al Power Hardware-in-the-Loop (PHIL), stanno assumendo un ruolo centrale per la sperimentazione di sistemi di controllo complessi operanti in scenari reali.

L'HIL introduce un'interazione bidirezionale tra un modello simulato in tempo reale e un dispositivo fisico sotto test (Device Under Test, DUT). L'evoluzione naturale di questa metodologia è il PHIL, che estende il concetto alla gestione di grandezze elettriche di potenza, permettendo così lo scambio reale di energia tra simulazione e hardware.

Il seguente elaborato si inserisce in questo contesto, con l'obiettivo di progettare, implementare e validare un sistema di controllo mediante architettura Power Hardware-in-the-Loop. Il lavoro analizza i principi teorici alla base del PHIL, esamina le principali metodologie di interfacciamento, presenta l'architettura sviluppata e discute i risultati sperimentali ottenuti confrontandoli con quanto ottenuto dalle simulazioni. Nell'elaborato verranno principalmente studiati due tipi di macchine elettriche: macchina sincrona a magneti superficiali lineare e macchina sincrona a riluttanza pura.

Nella prima parte verranno quindi analizzati gli ambienti di simulazione utilizzati e la costruzione dei modelli analitici delle due macchine all'interno del software PLECS (utile alla simulazione di azionamenti elettrici). Successivamente i modelli saranno confrontati con modelli a parametri concentrati disponibili all'interno delle librerie o costruiti appositamente per la verifica. Una particolare eccezione sarà fatta per la macchina a riluttanza pura perché ne verranno studiate e modellate anche le non linearità legate alla saturazione magnetica tramite l'utilizzo di software ad elementi finiti come FEMM.

La seconda parte avrà come obiettivo quello di dimostrare come l'approccio PHIL rappresenti uno strumento efficace, flessibile e sicuro per lo sviluppo e la validazione di sistemi di controllo avanzati, contribuendo a ridurre i tempi e i costi di progettazione e aumentando l'affidabilità delle soluzioni implementate. Saranno descritti e mostrati gli apparati sperimentali utilizzati ed i risultati dei modelli precedentemente studiati facendo un confronto per verificare l'affidabilità di questa architettura e l'onere computazionale che è richiesto.

1.1 PLECS

I modelli di macchina sviluppati sono stati implementati su una piattaforma hardware dedicata all'HIL programmabile mediate il software commerciale PLECS sviluppato da Plexim GmbH. Citando letteralmente il sito del produttore, il software PLECS viene descritto come*: "the tool of choice for high-speed simulations of power electronic systems"* [1].

L'idea alla base di PLECS è infatti quella di creare un programma di modellistica e simulazione specificatamente progettato per il mondo elettrico/elettronico, riconducendo il processo di modellistica il più possibile ad uno schema circuitale. La libreria inclusa nel software consente di modellare tutti gli aspetti dei sistemi di power elettronics e del loro controllo. Troviamo infatti componenti essenziali per il mondo elettrico ma anche magnetico, termico e meccanico, il tutto presentato tramite una finestra drag and drop ed un editor schematico.

La libreria più fornita è ovviamente quella elettrica: PLECS offre una vasta gamma di componenti a partire da quelli passivi fino ad una serie di circuiti di potenza basati su componenti ideali ma anche in grado di includere specifiche di produzione di componenti reali in modo da simulare correttamente elementi parassiti e i fenomeni termici. Inoltre, è presente una libreria di modelli di macchine elettriche con vari gradi di dettaglio e tutti interamente modificabili. Sono presenti anche componenti non lineari ed una serie di modelli di trasformatori pronti all'uso.

Anche per quanto riguarda la modellistica termica e magnetica viene riproposto lo schema circuitale tramite l'utilizzo di circuiti equivalenti. Sono inoltre disponibili componenti meccanici progettati per interagire facilmente con i modelli delle macchine elettriche in modo da modellare e simulare un sistema di controllo completo. Infine, grazie all'utilizzo di blocchi appositamente progettati per il controllo dei convertitori di potenza e soprattutto grazie al blocco C-Script (che consente di implementare direttamente in PLECS degli algoritmi di controllo in ANSI-C) è possibile realizzare facilmente qualunque schema di controllo digitale.

Il software è disponibile in due versioni: "PLECS Blockset" e "PLECS Standalone". Entrambe presentano la stessa libreria con componenti compatibili tra loro. La versione "Blockset" è integrabile con Simulink/Matlab e consente di usare i modelli e i componenti di PLECS con il solver, i blocchi di controllo e lo Script Editor di Simulink. Al contrario, la versione "Standalone" si presenta come una piattaforma di simulazione indipendente con un solver proprietario e ottimizzato. Durante lo svolgimento di questa tesi è stata utilizzata la versione "Blockset". Plexim GmbH produce anche un simulatore real-time chiamato "RT Box" che può essere programmato e utilizzato direttamente da PLECS tramite l'utilizzo di un'apposita libreria facilmente integrabile con le componenti e i modelli già citati.

1.2 Architettura Hardware in the Loop e Power Hardware in the Loop

L'architettura Power Hardware in the Loop (PHIL) ha origine dal concetto di Hardware in the Loop, con il quale si indica lo scambio di segnali in loop fra un sistema fisico reale sotto test (detto *device under test,* DUT) e un modello simulato capace di emulare il comportamento di un sistema fisico

qualsiasi. Questo scambio di segnali I/O analogici e digitali permette di ricreare le condizioni operative dell'apparecchio sotto test.

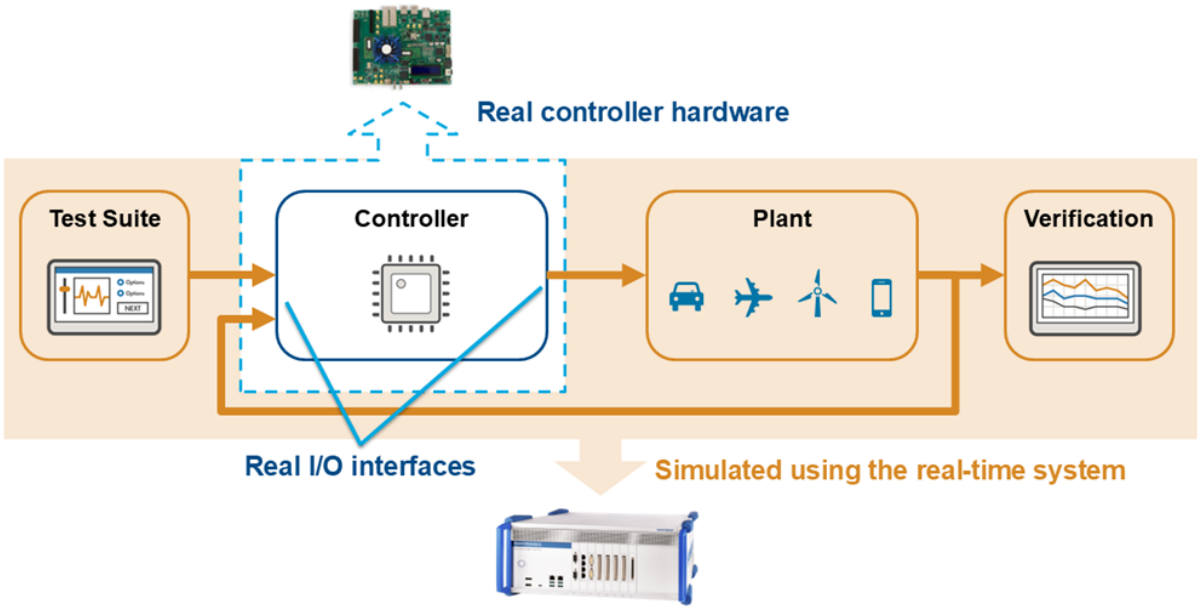


Figura 1: schema semplificativo HIL

Il funzionamento del PHIL, invece, sostituisce parte del sistema simulato con un'interfaccia di potenza che permette di avere uno scambio reale di energia con il DUT (vedi Fig. 2). Quindi a differenza del HIL che gestisce solo segnali a bassa potenza, il PHIL utilizza amplificatori di potenza per emulare il comportamento di un sistema complesso, permettendo di testare componenti ad alta tensione e corrente, come inverter o motori elettrici, in condizioni operative realistiche e sicure.

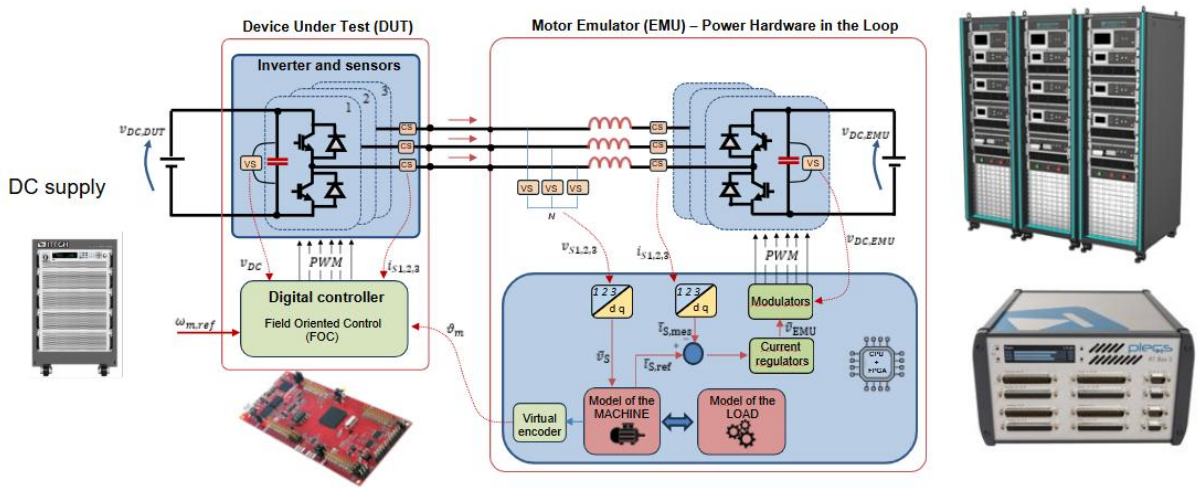


Figura 2: schema semplificativo dell'architettura PHIL

La conversione del segnale simulato a quello di potenza è mediata da due componenti chiave: l'interfaccia di potenza e il dispositivo di accoppiamento che verranno approfonditi in seguito.

1.2.1 Struttura PHIL

Per testare un dispositivo di elettronica di potenza, è necessario un apparecchio che trasformi i segnali provenienti dal modello matematico in segnali di potenza. Riprodurre un segnale elettrico variabile nel tempo come quello in uscita dal modello matematico non è semplice, ma può essere ottenuto come la media di un'onda quadra in un determinato periodo. Ciò è svolto da un convertitore statico di potenza DC-AC (inverter). Quest'ultimo trasforma una tensione continua (DC), che funge da alimentazione primaria per il processo di conversione, in una tensione alternata (AC). Il cuore dell'inverter è rappresentato dallo stadio di commutazione, composto da transistor o altri dispositivi (come IGBT o MOSFET), che hanno il compito di aprire e chiudere i circuiti in modo rapido e controllato.

Nella maggior parte dei casi, i convertitori sono pilotati da segnali generati da una logica di modulazione a larghezza di impulso (PWM). Questo approccio consente di variare la durata degli impulsi di tensione nel tempo, mimando così la forma e le caratteristiche volute dal modello. Dovendo mantenere il passo con il tempo reale, la finestra di tempo su cui mediare gli impulsi deve essere più piccola rispetto a quella usata dal DUT. Convertitori e transistor di nuova generazione, sempre più veloci nell'aprire e chiudere il canale, rendono possibili frequenze di commutazione elevate adatte allo scopo. Tuttavia, il segnale generato dall'operazione di commutazione contiene spesso un numero significativo di armoniche, dovute alla natura non perfettamente sinusoidale dell'onda prodotta.

Interfaccia di potenza

L'interfaccia di potenza costituisce il cuore del collegamento tra la simulazione e il dispositivo reale. Il suo compito principale è quello di tradurre i segnali calcolati dal simulatore in grandezze elettriche fisiche, in modo che il dispositivo in prova (DUT) percepisca condizioni del tutto analoghe a quelle di un sistema reale (vedi Fig. 2). Le sue funzioni principali sono:

- *Generazione di tensioni o correnti*: l'interfaccia prende i riferimenti dal simulatore e li riproduce con un amplificatore di potenza.
- *Misura delle risposte del DUT*: correnti e tensioni generate dal dispositivo vengono rilevate da sensori di precisione e inviate in tempo reale al simulatore.
- *Protezione e sicurezza*: in caso di sovracorrente, guasti o instabilità, l'interfaccia deve isolare rapidamente il DUT e preservare sia l'hardware reale che il simulatore.

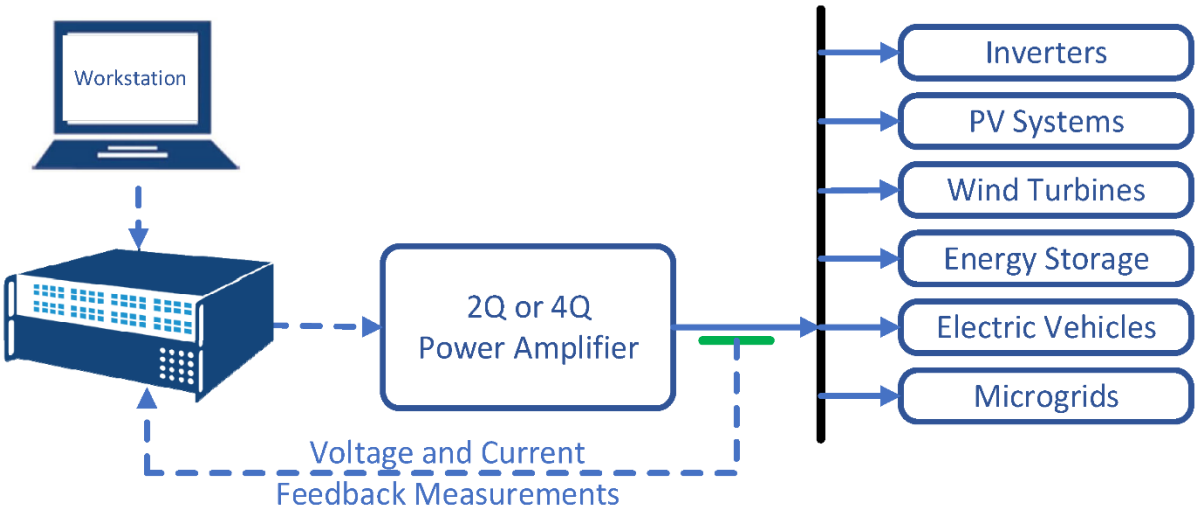


Figura 3:Architettura PHIL con amplificatore di potenza

Dispositivi di accoppiamento

Il dispositivo di accoppiamento svolge un ruolo complementare a quello dell'interfaccia di potenza in quanto è essenziale per garantire che i flussi di potenza siano trasmessi efficacemente tra l'ambiente simulato e il DUT. Nel caso di una configurazione PHIL, se il DUT fosse direttamente connesso all'emulatore, non si avrebbe nessun tipo di controllo sulle correnti che si scambiano. Quest'ultimo è solitamente una resistenza in serie ad un induttore. Ciò fa sì che si possa determinare il flusso di corrente sapendo le tensioni applicate. Inoltre, l'induttore svolge anche un'azione filtrante per le correnti che altrimenti sarebbero più distorte. Il dispositivo di accoppiamento ha quindi il compito di adattare dinamicamente i segnali di potenza dell'emulatore alle condizioni operative reali del DUT e deve assicurare la stabilità del ciclo.

A seconda delle applicazioni, il dispositivo di accoppiamento può assumere configurazioni diverse, ma in generale i suoi compiti sono:

- *Adattamento di livello*: in molti casi i segnali di uscita dell'interfaccia devono essere adattati in tensione, corrente o frequenza per poter essere applicati in sicurezza al DUT.
- *Isolamento galvanico*: garantisce la separazione elettrica tra il simulatore e il dispositivo reale, riducendo il rischio di cortocircuiti, guasti a cascata o ritorni di potenza indesiderati.
- *Riduzione del rumore e dei disturbi*: tramite filtri attivi o passivi il dispositivo di accoppiamento limita armoniche e transitori che potrebbero compromettere sia la stabilità della simulazione sia l'integrità del DUT.
- *Protezione bidirezionale*: agisce come barriera di sicurezza, in modo che eventuali anomalie dal lato reale (ad esempio cortocircuiti o fault transitori) non danneggino il simulatore e viceversa.

La combinazione di interfaccia di potenza e dispositivo di accoppiamento consente al PHIL di operare in modo affidabile, garantendo che lo scambio di energia tra simulazione e dispositivo reale sia fedele, stabile e sicuro. In questo modo è possibile testare apparecchiature ad alta potenza in scenari realistici senza esporre il sistema complessivo ai rischi di un impianto reale.

Capitolo 2

Modellazione delle Macchine Sincrone e Sistemi di Controllo in Orientamento di Campo

In questo capitolo vengono analizzati e sviluppati i modelli matematici e gli schemi di controllo necessari per la rappresentazione e la regolazione di macchine sincrone impiegate in applicazioni ad alta dinamica. L'obiettivo è fornire una descrizione completa degli azionamenti e delle strategie di controllo utilizzate, evidenziando le differenze strutturali e di comportamento tra le diverse tipologie di macchine considerate. In particolare, verranno approfonditi due modelli distinti: una macchina sincrona a magneti superficiali (Surface Permanent Magnet – SPM), modellata in condizioni di linearità magnetica dei materiali, e una macchina sincrona a riluttanza pura, per la quale verranno analizzate in dettaglio le non linearità dovute alla saturazione del circuito magnetico.

La definizione accurata dei modelli è un passaggio fondamentale per la validazione degli algoritmi di controllo, poiché consente di riprodurre il comportamento dinamico delle macchine in condizioni reali di funzionamento tenendo conto delle interazioni tra corrente, flusso e coppia elettromagnetica. Nel caso della macchina SPM, verrà adottata una modellazione lineare basata sulla classica rappresentazione nel sistema di riferimento d-q, nella quale i parametri magnetici sono assunti costanti e l'accoppiamento incrociato tra gli assi è minimo o trascurabile. Questo tipo di modello permette un'analisi chiara e immediata delle prestazioni del controllo, risultando particolarmente adatto per studi nei quali la semplicità del modello è funzionale alla progettazione del regolatore.

Di natura più complessa è invece la modellazione della macchina sincrona a riluttanza pura. A differenza del caso lineare, in questa configurazione il comportamento magnetico del materiale introduce effetti significativi di saturazione e anisotropia, che si traducono in una variazione non lineare delle induttanze al variare della corrente. Ciò comporta una maggiore difficoltà nella predizione delle prestazioni del sistema e richiede l'adozione di modelli capaci di catturare tali fenomeni, al fine di valutare con precisione la coppia prodotta e la stabilità del controllo. Nel capitolo verranno quindi discusse le tecniche utilizzate per includere le non linearità nella modellazione, illustrando le differenze rispetto al caso lineare.

Oltre alla costruzione dei modelli, verranno presentati i controlli di velocità impiegati nei due azionamenti. L'analisi comprenderà sia la struttura generale dell'anello di regolazione, sia i criteri di progettazione del regolatore e le considerazioni relative alle proprietà dinamiche delle due macchine.

2.1 Modello di un azionamento elettrico basato su una macchina SPM lineare

Per poter effettuare una sperimentazione accurata della macchina elettrica di interesse, è necessario disporre preliminarmente di un modello matematico adeguato che ne descriva le dinamiche fondamentali. Senza un modello di riferimento non sarebbe infatti possibile né implementare correttamente l'interfaccia PHIL, né analizzare i fenomeni fisici e le prestazioni della macchina durante la simulazione.

La costruzione del modello matematico deve rispettare le leggi fondamentali dell'elettrotecnica e della fisica dei sistemi dinamici, come la legge di Ohm, le leggi di Kirchhoff e le equazioni elettromeccaniche che descrivono la conversione di energia tra parte elettrica e parte meccanica. A seconda della tipologia di macchina elettrica (sincrona, asincrona, a magneti permanenti, a corrente continua, ecc.) il modello dovrà includere le equazioni differenziali che regolano il flusso di corrente, di tensione e di flusso magnetico, oltre alle relazioni meccaniche legate a coppia, velocità angolare e inerzia.

Il primo modello studiato, per semplicità, è quello di una macchina sincrona lineare, isotropa a magneti superficiali.

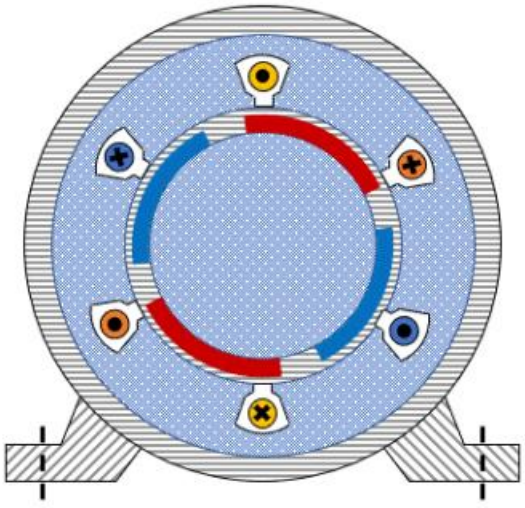


Figura 4: sezione macchina SPM

È stato poi effettuato un ulteriore confronto, atto a verificare la veridicità del modello, con un blocchetto motore già esistente nelle librerie di PLECS. Il motore di riferimento impiegato per verificare i modelli è il prodotto commerciale BTSS 1524 sviluppato da TEM Electric Motors. La scheda tecnica della macchina è riportata nella figura seguente.

DATI DEL MOTORE	SIMBOLO	VALORE	UNITÀ DI MISURA
COPPIA ROTORE BLOCCATO ΔT 100°C IN ARIA	C_m	0.6	Nm
COPPIA DI PICCO	$C_{m,max}$	2.25	Nm
VELOCITA' NOMINALE	Nn	1500	rpm
VELOCITA' MAX	Nn_{max}	5000	rpm
POTENZA NOMINALE IN ARIA ΔT 100°C	P_n	95	W
CORRENTE A VELOCITA' NOM. ΔT 100°C IN ARIA	I_n	2.85	Arms
CORRENTE DI PICCO	I_{max}	10.8	Arms
MOMENTO D' INERZIA ROTORICO	J	0.380	$kg \cdot m^2 \cdot 10^{-4}$
F.E.M CONCATENATA	K_e	0.123	V·s
COSTANTE DI COPPIA	K_T	0.21	Nm/A
RESISTENZA CONCATENATA	R_w	0.74	Ω
INDUTTANZA CONCATENATA	L_w	1.4	mH
FREQUENZA	f	100	Hz
RENDIMENTO A POTENZA NOM. ΔT 100°C	η	93	%
N° POLI	p	8	

Figura 5: specifiche motore TEM BTSS 1524

Per descrivere il funzionamento della macchina SPM si può partire dalle equazioni generali di bilancio delle tensioni v_a, v_b, v_c, che con la convenzione di segno degli utilizzatori risultano:

$$v_a = Ri_a + \frac{d\varphi_a}{dt} \quad (1)$$

$$v_b = Ri_b + \frac{d\varphi_b}{dt} \quad (2)$$

$$v_c = Ri_c + \frac{d\varphi_c}{dt} \quad (3)$$

$$\varphi_a = L_s i_a + \phi_e \cos\vartheta \quad (4)$$

$$\varphi_b = L_s i_b + \phi_e \cos\left(\vartheta - \frac{2}{3}\pi\right) \quad (5)$$

$$\varphi_c = L_s i_c + \phi_e \cos\left(\vartheta - \frac{4}{3}\pi\right) \quad (6)$$

Dove i_a, i_b, i_c sono le correnti che percorrono le tre fasi, $\varphi_a, \varphi_b, \varphi_c$ sono i flussi magnetici concatenati con ciascuna fase ed R è la resistenza di fase, che si suppone uguale per le tre fasi. Nelle equazioni dei flussi compare L_s, cioè il coefficiente di autoinduzione dell'avvolgimento statorico. Tale parametro non coincide con l'induttanza sincrona, che viene definita solo dopo la trasformazione nel riferimento $d-q$ e tiene conto degli effetti combinati delle autoinduttanze e delle mutue tra le fasi. ϕ_e

è il valore di picco del flusso concatenato con un avvolgimento statorico, dovuto al campo di eccitazione prodotto dai magneti del rotore. ϕ_e viene poi moltiplicato per il coseno dell'angolo del flusso rispetto alla posizione delle tre fasi statoriche. Per semplificare l'utilizzo di queste equazioni nei sistemi di controllo, si passa, con l'utilizzo dei vettori di spazio, ad un modello bifase (trasformazione di Clarke). Successivamente, si descrive il modello bifase in un sistema di riferimento rotante d-q in modo sincrono al rotore e con asse d orientato secondo la direzione del campo di eccitazione. Tale sistema risulta essere orientato di un angolo ϑ rispetto a quello statorico; dunque, per esprimere le grandezze elettriche in un riferimento solidale con il rotore è sufficiente applicare al modello bifase una trasformazione di rotazione, nota come trasformata di Park. Prima della trasformata di Park si applica la trasformata di Clarke, che consente di passare dal sistema a tre fasi (abc) al sistema bifase stazionario $(\alpha\beta)$. Successivamente, la trasformata di Park ruota tale sistema di un angolo ϑ, ottenendo le grandezze nel riferimento rotante $(d-q)$.

Trasformata di Clarke

$$v_\alpha = Ri_\alpha + \frac{d\varphi_\alpha}{dt}$$

$$v_\beta = Ri_\beta + \frac{d\varphi_\beta}{dt} \qquad (7)$$

$$\varphi_\alpha = L_s i_\alpha + \phi_{e\,\alpha}$$

$$\varphi_\beta = L_s i_\beta + \phi_{e\,\beta} \qquad (8)$$

Trasformata di Park (forma vettoriale)

$$\bar{v} = R\bar{\imath} + \frac{d\bar{\varphi}}{dt} + j\omega\bar{\varphi} \qquad (9)$$

$$\bar{\varphi} = L_s\bar{\imath} + \phi_e \qquad (10)$$

in cui compare la velocità del rotore ω in radianti elettrici al secondo, ovvero $\omega = p\omega_m$ con p numero di poli del motore. Sostituendo nella prima la seconda ed esplicitandole nelle componenti d-q, diventano:

$$v_d = Ri_d + L_s\frac{di_d}{dt} - \omega L_s i_q \qquad (11)$$

$$v_q = Ri_q + L_s\frac{di_q}{dt} + \omega L_s i_d + \omega\phi_e \qquad (12)$$

Tramite un bilancio di potenze, è possibile ricavare la coppia meccanica erogata all'albero:

$$C_m = \frac{3}{2}pi_q\phi_e \qquad (13)$$

Spesso le specifiche del motore fornite dal costruttore non comprendono tutti i dati necessari per la modellazione dello stesso, quelli mancanti possono essere ricavati. Ad esempio, i dati di Fig. 5 del motore SPM di riferimento non includono direttamente Rs, Ls e ϕ_e. L'induttanza sincrona può essere ricavata dall'induttanza concatenata riportata nelle specifiche, che è misurata ai capi di due fasi. Immaginando di alimentare, con un generatore di tensione, solo due delle tre fasi del motore, il flusso creato sarà espresso dalle equazioni *(1), (2), (3)*. Prendiamo, ad esempio, la fase a e la b e applichiamo

una tensione ai loro capi come appena descritto, la corrente che circolerà nella fase a sarà uguale e opposta a quella nella fase b ($i_a = -i_b$). Il flusso, senza la presenza del rotore, sarà espresso da:

$$\varphi_a - \varphi_b = L_s i_a - L_s i_b = 2L_s i_a = L_w i_a \quad (14)$$

Dunque, l'induttanza sincrona presente nelle equazioni del motore risulta essere

$$L_s = \frac{L_w}{2} \quad (15)$$

Analogo ragionamento può essere fatto per la resistenza concatenata

$$R_s = \frac{R_w}{2} \quad (16)$$

Il flusso concatenato ϕ_e è ricavato dalla costante di tensione. Tale costante è definita come il rapporto fra la coppia e la corrente di asse q

$$K_e = \frac{C_m}{i_q} = \frac{3}{2} p \phi_e \quad (17)$$

Manovrando l'equazione e considerando la corrente efficace, è possibile ottenere il flusso concatenato

$$\phi_e = \frac{2}{3p} \frac{K_e}{\sqrt{2}} = 0.0247\ Wb \quad (18)$$

Per poter implementare il modello matematico della macchina all'interno di software di simulazione come PLECS o Simulink si parte dalle equazioni delle correnti ricavate dalle eq. *(11), (12)* nel dominio di Laplace:

$$i_d = \frac{V_d + \omega_e L_s i_q}{sL_s + R_s} \quad (19)$$

$$i_q = \frac{V_q - \omega_e L_s i_d - \omega_e \phi_e}{sL_s + R_s} \quad (20)$$

Le equazioni *(11), (12)* sono funzione delle tensioni imposte dal convertitore, misurate e trasformate nel sistema di assi d-q. Per la costruzione dello schema a blocchi utilizzeremo quindi le equazioni *(19), (20)* e la seguente equazione della dinamica dell'albero dalla quale è possibile ricavare la velocità ω_m

$$C_m - C_r = J_{tot} \frac{d\omega_m}{dt} \rightarrow \omega_m = \frac{C_m - C_r}{sJ_{tot}} \quad (21)$$

Nella figura seguente è quindi possibile vedere come realizzare lo schema blocchi di un motore sincrono isotropo [2]:

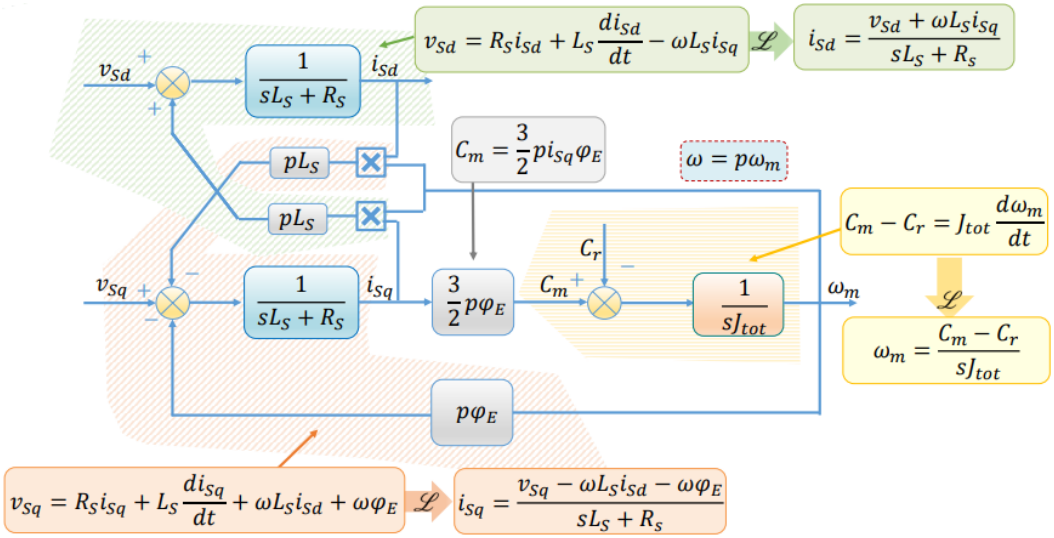


Figura 6: schema a blocchi motore sincrono isotropo

L'equazione di coppia studiata in precedenza *(13)* è dipendente solo dalla corrente di asse q; quindi, lo schema blocchi implementabile per il controllo può essere visto nel seguente modo:

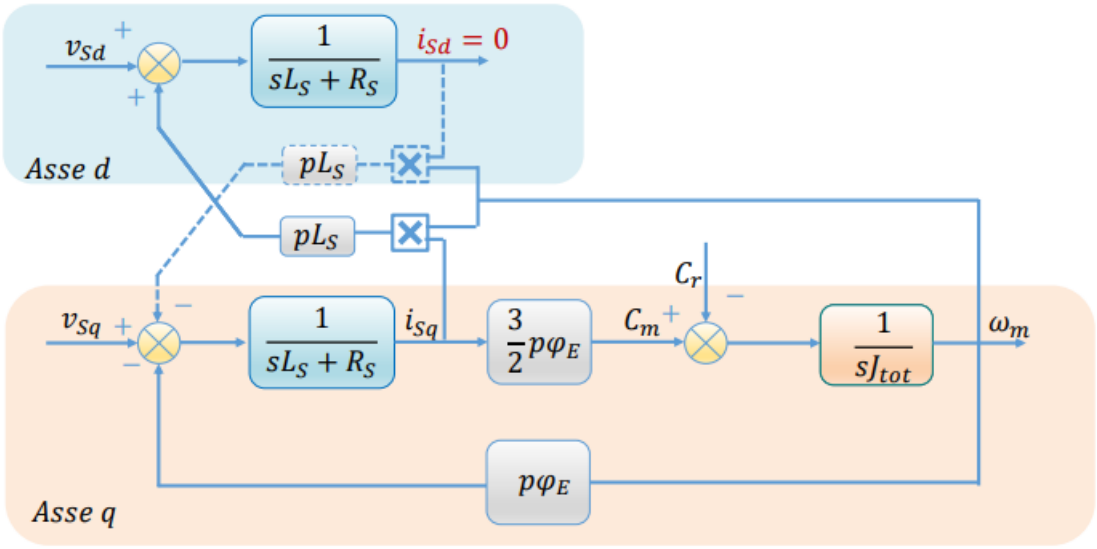


Figura 7: schema a blocchi con asse q indipendente da asse d

Di conseguenza il principio di funzionamento di un controllo ad orientamento di campo è quello di iniettare nelle tre fasi di statore delle correnti in grado di generare un campo magnetico che ruota sempre al sincronismo con il rotore. Possiamo quindi vedere il funzionamento del sistema nel seguente modo:

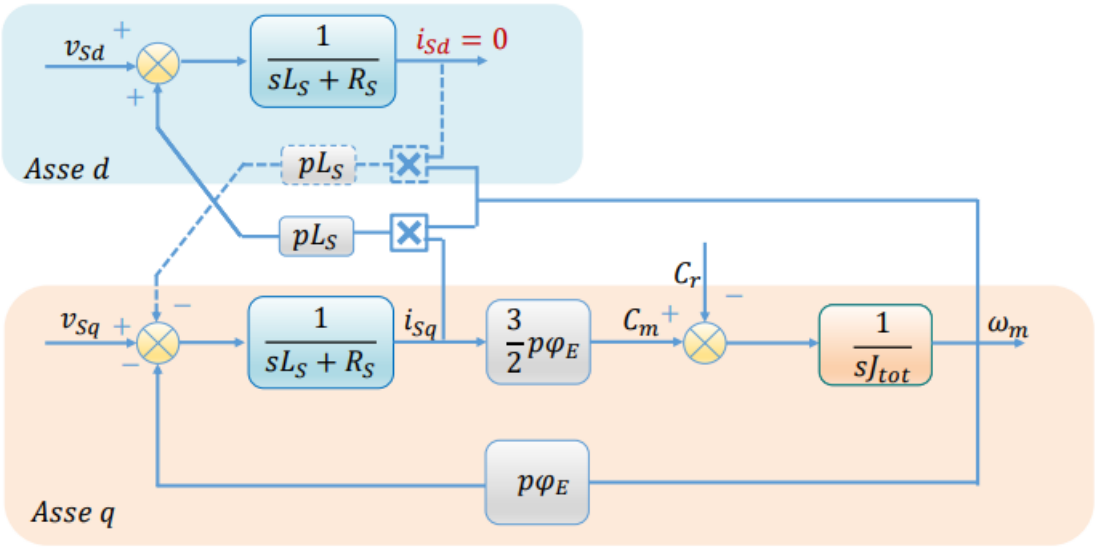


Figura 8: schema a blocchi con asse q indipendente da asse d

A parità di corrente assorbita, la coppia è massimizzata se $i_{sd} = 0$. Per riuscire in questo è necessario conoscere, istante per istante, la posizione del rotore. Dopo le compensazioni delle fem dinamiche, l'asse d presenta una funzione di trasferimento uguale a quella dell'asse q. In seguito alla compensazione delle fem dinamiche i due assi sono indipendenti e la taratura dei regolatori di corrente può essere identica e fatta secondo il criterio di compensazione polo-zero.

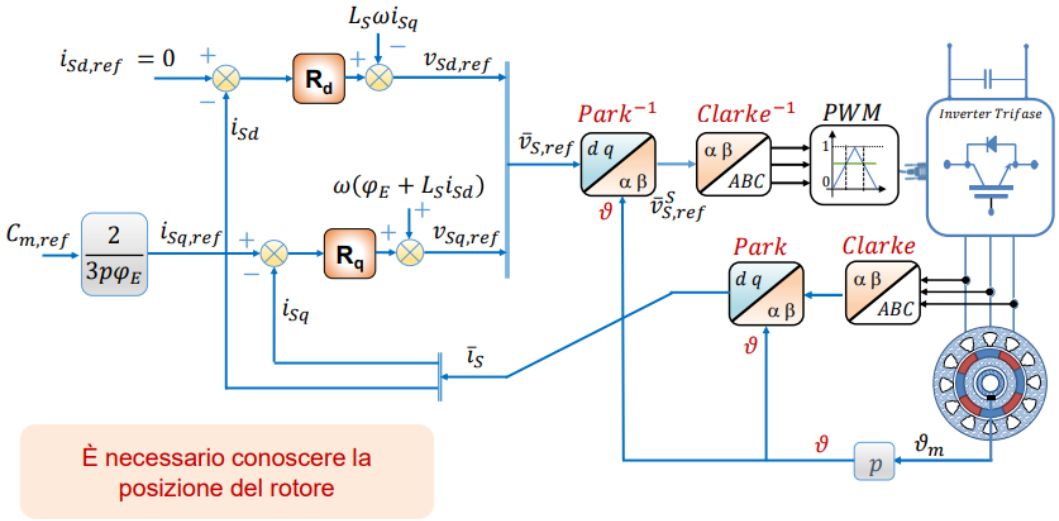


Figura 9: controllo di coppia

L'asse d ha un ruolo fondamentale durante il funzionamento ad alta velocità mentre a bassa velocità è posto uguale a zero. L'impiego di una corrente i_{sd} negativa permette, a parità di corrente presente nelle fasi di statore e di tensione all'inverter, di raggiungere velocità di rotazione superiori; si parla di *indebolimento di campo o deflussaggio.*

La coppia è controllabile mediante i_q, quindi per chiudere anche l'anello di velocità è sufficiente aggiungere un ulteriore regolatore che confronta la velocità misurata con quella di riferimento andando ad agire sul riferimento di corrente.

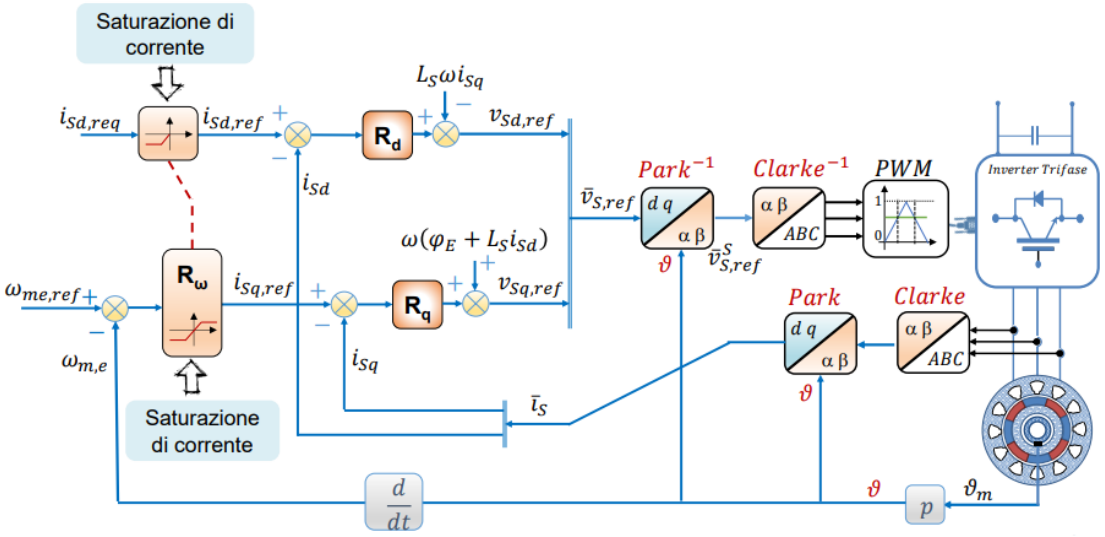


Figura 10:controllo di velocità

2.1.1 Prestazioni limite dell'azionamento

Le prestazioni limite di un azionamento definiscono i massimi livelli di coppia, corrente e dinamica che il sistema può raggiungere. Considerarle è essenziale per capire fino a che punto l'azionamento può seguire velocemente una variazione di velocità senza superare i vincoli imposti dalla macchina, inverter e controllo.

Limite in corrente

In regime simmetrico e sinusoidale la traiettoria del vettore corrente nel piano complesso $i_{s\alpha} - i_{s\beta}$ è circolare. I riferimenti di corrente devono essere compatibili con i limiti termici dell'inverter e della macchina. Il limite in corrente $i_{s\alpha} - i_{s\beta}$ nel piano è descritto da una circonferenza il cui raggio è il valore di picco della corrente di fase massima ammissibile.

$$\begin{cases} |\bar{i}_s^s| = |\bar{i}_s| \leq I_{Max} \\ i_{sd}^2 + i_{sq}^2 \leq I_{Max}^2 \end{cases} \qquad (22)$$

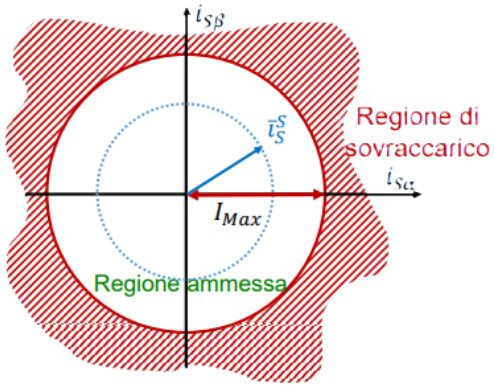


Figura 11:limite di corrente

Limite in tensione

I regolatori di corrente producono in uscita un riferimento di tensione $\bar{v}_s$ che può essere applicato solo se la tensione continua di alimentazione dell'inverter E_{DC} è sufficientemente elevata, ovvero $\frac{E_{DC}}{\sqrt{3}}$. In regime simmetrico e sinusoidale, il vettore tensione di riferimento nel piano $v_{s\alpha} - v_{s\beta}$ percorre una traiettoria circolare.

$$\begin{cases} |\bar{v}_s^s| = |\bar{v}_s| \leq V_{Max} \\ v_{sd}^2 + v_{sq}^2 \leq V_{Max}^2 \end{cases} \quad (23)$$

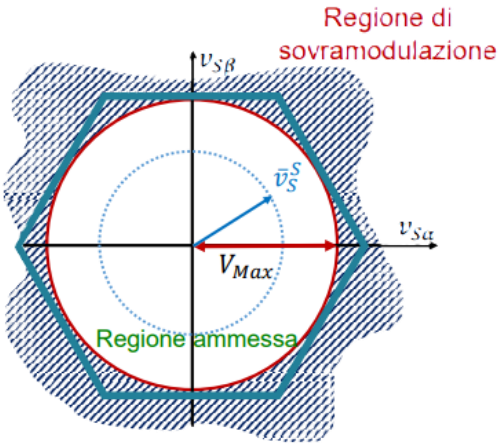


Figura 12: limite di tensione

L'espressione della coppia è la più facile da descrivere in quanto non dipende dalla componete reale del vettore di spazio della corrente (i_{sd}). Per produrre la coppia con le minime perdite per effetto Joule è conveniente imporre $i_{sd} = 0$.

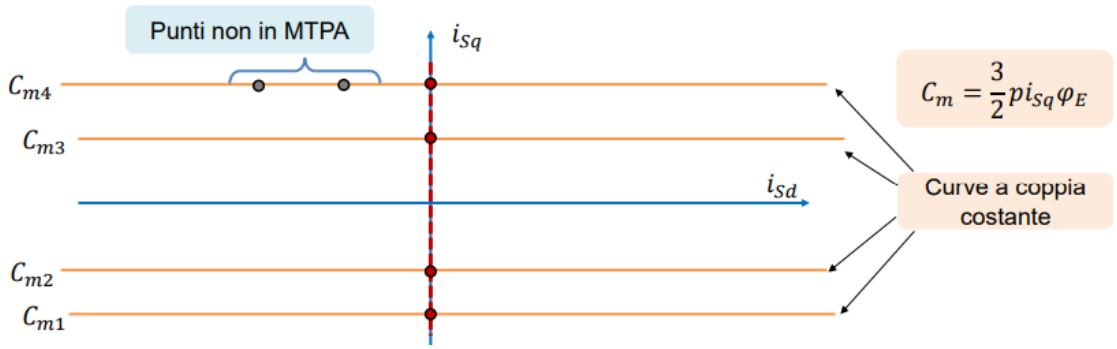


Figura 13:curva di coppia

Componendo tutte le curve (coppia, limite tensione e limite corrente) si ha che i punti di funzionamento accettabili sono quelli in grado di fornire la coppia richiesta dal carico rispettando i limiti di tensione e di corrente.

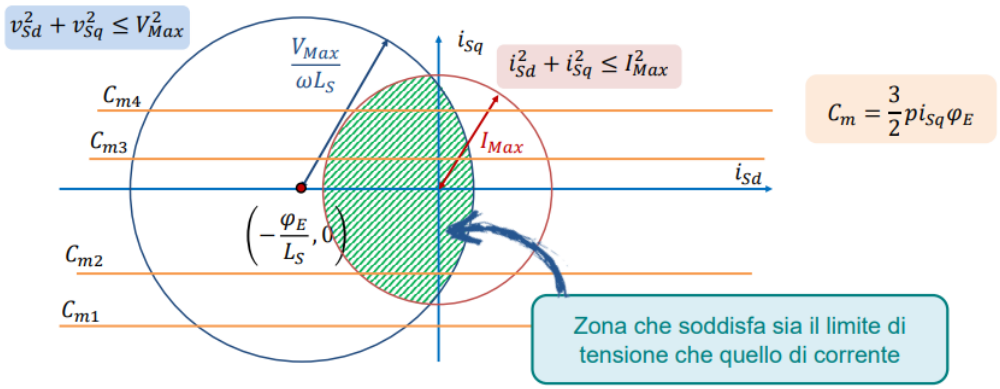


Figura 14:prestazioni limite dell'azionamento

Se la i_{sd} è mantenuta uguale a zero anche oltre la velocità base la coppia massima erogabile diminuisce fino a diventare nulla alla velocità massima.

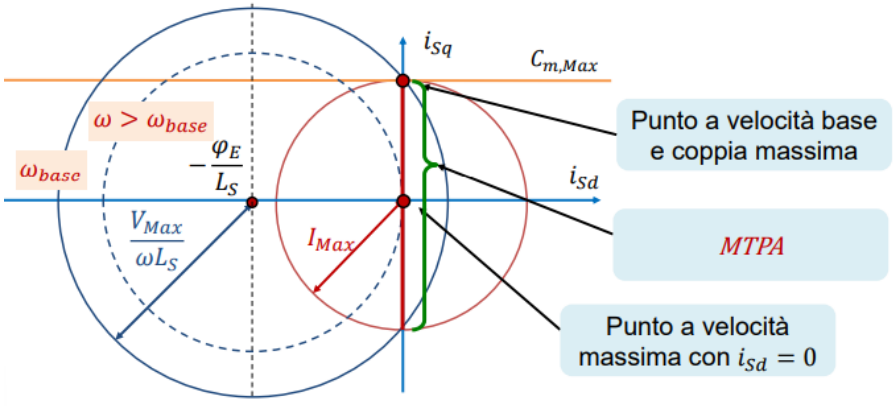


Figura 15:condizioni di MTPA

Spiegati quelli che sono i limiti dell'azionamento risulta ora più chiaro il motivo per il quale nel controllo di velocità di figura 10 sono presenti due regolatori saturati; questo perché la componente i_{sq} della corrente è collegata alla produzione di coppia mentre i_{sd} è legata alla capacità di operare in deflussaggio. Una possibile strategia di limitazione è quella che da priorità alla componente i_{sq} della corrente in quanto ad essa è collegata la produzione della coppia. In tal caso, qualora la richiesta di corrente fosse non compatibile con il limite in corrente, si privilegia la componente i_{sq} e la i_{sd} viene saturata allo scopo di riportare il vettore corrente dentro al cerchio limite.

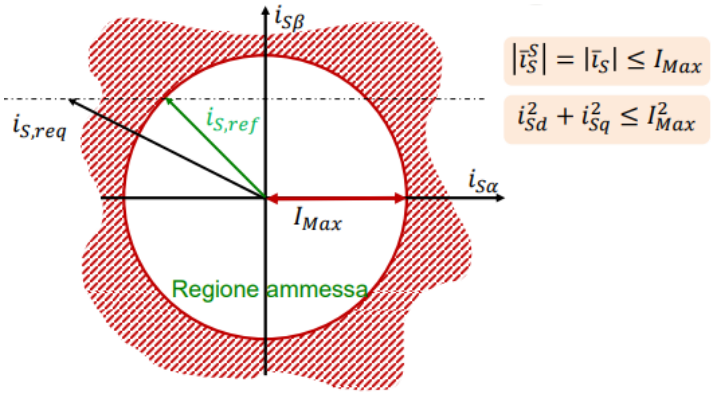


Figura 16: regolazione limite di corrente

2.2 Azionamento Motore a Riluttanza Pura

Nelle applicazioni ad alta velocità, le macchine a magneti permanenti (SPM) sono indubbiamente considerate le più performanti in termini di efficienza, densità di coppia, densità di potenza e ampiezza del campo di velocità. Purtroppo, però, il prezzo delle terre rare e dei conseguenti magneti utilizzati nei SPM è recentemente aumentato ed è soggetto a fluttuazioni imprevedibili; inoltre, il loro utilizzo comporta un approccio poco favorevole al riciclaggio dei materiali a fine vita e la produzione di elementi a terre rare richiede un elevato consumo di materiali ed energia ed è associata a notevoli oneri ambientali.

Una possibile soluzione a questo problema possono essere i motori a riluttanza pura o riluttanza sincrona (SyncRel) che non impiegano magneti permanenti e sono utilizzati ugualmente in applicazioni ad alta velocità, al momento però presentano criticità dovute alla poca robustezza del rotore. Questi motori risultano però molto interessanti per via delle loro prestazioni in assenza di magneti e la ricerca si sta sempre più immergendo nello studio del miglioramento delle prestazioni e della realizzazione. Ad esempio, negli ultimi anni, con l'avvento del 3D printing si sta cercando di migliorare le prestazioni di questi motori andando a stampare e ricostruire parti di motore con metodi molto particolari che permettono di mantenere le stesse proprietà magnetiche ed elettriche e modificare proprietà meccaniche e materiali di esso [3].

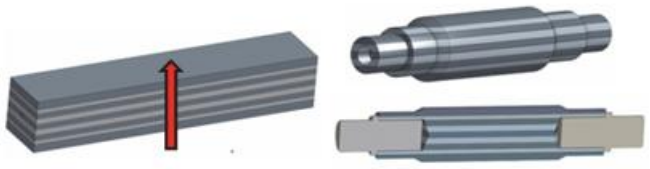


Figura 17: realizzazione 3D printing rotore a riluttanza

Il modello spiegato in precedenza (brushless isotropo) è semplificato perché lineare, ovvero, vengono trascurati alcuni fenomeni fisici che, in condizioni operative reali, influenzano in modo significativo la risposta del sistema. Ad esempio, la saturazione magnetica dei materiali ferromagnetici, le variazioni di riluttanza al variare della posizione rotorica, le perdite ferromagnetiche non lineari e le non linearità meccaniche dovute ad attriti e coppie di carico variabili con la velocità. Tali effetti diventano particolarmente rilevanti in presenza di elevate correnti di fase, alte velocità o transitori dinamici rapidi, situazioni in cui il modello lineare non è più in grado di garantire un'accurata rappresentazione del

comportamento del motore. Per questo motivo, risulta necessario sviluppare una modellazione non lineare capace di riprodurre, in modo più fedele, le interazioni tra il circuito elettrico, il campo magnetico e la dinamica meccanica del sistema.

2.2.1 Modelli di macchine sincrone non lineari

Per considerare quelli che sono gli effetti della saturazione sul nostro modello ci sono varie possibilità che dipendono anche da quella che poi sarà la potenza di calcolo disponibile; sono stati considerati principalmente due modi per la risoluzione del problema più un terzo per effettuare una verifica finale:

- ***Modello 4D***

  Partendo dalle equazioni generali di bilancio delle tensioni per una generica fase

$$v_a = R_a i_a + \frac{d\varphi_a}{dt} \tag{24}$$

  si deve tenere conto che $\varphi_a = \varphi_a(i_a, i_b, i_c, \vartheta)$, ovvero che il flusso di ogni fase dipende dalle tre correnti e dalla posizione angolare del rotore. Questo comporterebbe uno studio matriciale molto complesso dove ogni singola derivata dei flussi deve essere calcolata rispetto le 4 variabili da cui dipende e per ogni valore di essi.

  Di seguito viene mostrato concettualmente lo sviluppo dell'equazione soprariportata considerando il contributo di tutte e tre le fasi.

$$\begin{aligned}
\frac{d\varphi_a}{dt}_{|i_a,i_b,i_c,\vartheta} &= \frac{\partial\varphi_a}{\partial i_a}_{|i_a,i_b,i_c,\vartheta}\frac{di_a}{dt} + \frac{\partial\varphi_a}{\partial i_b}_{|i_a,i_b,i_c,\vartheta}\frac{di_b}{dt} + \frac{\partial\varphi_a}{\partial i_c}_{|i_a,i_b,i_c,\vartheta}\frac{di_c}{dt} + \frac{\partial\varphi_a}{\partial\vartheta}\frac{d\vartheta}{dt} \\
\frac{d\varphi_b}{dt}_{|i_a,i_b,i_c,\vartheta} &= \frac{\partial\varphi_b}{\partial i_a}_{|i_a,i_b,i_c,\vartheta}\frac{di_a}{dt} + \frac{\partial\varphi_b}{\partial i_b}_{|i_a,i_b,i_c,\vartheta}\frac{di_b}{dt} + \frac{\partial\varphi_c}{\partial i_c}_{|i_a,i_b,i_c,\vartheta}\frac{di_c}{dt} + \frac{\partial\varphi_b}{\partial\vartheta}\frac{d\vartheta}{dt} \\
\frac{d\varphi_c}{dt}_{|i_a,i_b,i_c,\vartheta} &= \frac{\partial\varphi_c}{\partial i_a}_{|i_a,i_b,i_c,\vartheta}\frac{di_a}{dt} + \frac{\partial\varphi_c}{\partial i_b}_{|i_a,i_b,i_c,\vartheta}\frac{di_b}{dt} + \frac{\partial\varphi_c}{\partial i_c}_{|i_a,i_b,i_c,\vartheta}\frac{di_c}{dt} + \frac{\partial\varphi_c}{\partial\vartheta}\frac{d\vartheta}{dt}
\end{aligned} \tag{25}$$

  Sostituendo la *(25)* nella *(24)* è ora possibile ottenere quanto segue

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{\partial\varphi_a}{\partial i_a} & \frac{\partial\varphi_a}{\partial i_b} & \frac{\partial\varphi_a}{\partial i_c} \\ \frac{\partial\varphi_b}{\partial i_a} & \frac{\partial\varphi_b}{\partial i_b} & \frac{\partial\varphi_b}{\partial i_c} \\ \frac{\partial\varphi_c}{\partial i_a} & \frac{\partial\varphi_c}{\partial i_b} & \frac{\partial\varphi_c}{\partial i_c} \end{bmatrix}\frac{d}{dt}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{\partial\varphi_a}{\partial\vartheta} \\ \frac{\partial\varphi_b}{\partial\vartheta} \\ \frac{\partial\varphi_c}{\partial\vartheta} \end{bmatrix}\omega \tag{26}$$

A questo punto sarà possibile ricavare il valore delle correnti; per semplicità la matrice delle derivate dei flussi verrà chiamata [A] e la matrice diagonale delle resistenze [R_s].

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \int [A]^{-1}\left(\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - R_s\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} \frac{\partial\varphi_a}{\partial\vartheta} \\ \frac{\partial\varphi_b}{\partial\vartheta} \\ \frac{\partial\varphi_c}{\partial\vartheta} \end{bmatrix}\omega\right)dt \tag{27}$$

Questo tipo di modellizzazione è molto accurata ma allo stesso tempo richiede una potenza di calcolo molto elevata per poter gestire l'onere computazionale richiesto; risulta quindi più comodo considerare il modello seguente che è molto simile al 4D ma richiede una potenza di calcolo minore.

- ***Modello 3D***

  Questo tipo di modellizzazione concettualmente sviluppa gli stessi passaggi del modello 4D con la differenza che le tre correnti di fase (i_a, i_b, i_c) verranno considerate utilizzando il riferimento d-q e di conseguenza perderemo la dipendenza da una variabile [4]. Partendo dall'equazione che descrive le tensioni di fase del motore:

$$\bar{V} = R\bar{\imath} + \frac{d\bar{\varphi}}{dt} + j\omega\bar{\varphi}$$

  È possibile ottenere le equazioni nei riferimenti d-q:

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = R_s \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \frac{d\varphi_{sd}}{dt} \\ \frac{d\varphi_{sq}}{dt} \end{bmatrix} + j\omega \begin{bmatrix} \varphi_{sd} \\ \varphi_{sq} \end{bmatrix} \tag{28}$$

  Tenendo conto che la matrice delle derivate dei flussi è uguale a:

$$\begin{aligned} \frac{d\varphi_{sd}}{dt} &= \frac{\partial\varphi_{sd}}{\partial i_{sd}}\frac{di_{sd}}{dt} + \frac{\partial\varphi_{sd}}{\partial i_{sq}}\frac{di_{sq}}{dt} + \frac{\partial\varphi_{sd}}{\partial\vartheta}\frac{d\theta}{dt} \\ \frac{d\varphi_{sq}}{dt} &= \frac{\partial\varphi_{sq}}{\partial i_{sd}}\frac{di_{sd}}{dt} + \frac{\partial\varphi_{sq}}{\partial i_{sq}}\frac{di_{sq}}{dt} + \frac{\partial\varphi_{sq}}{\partial\vartheta}\frac{d\theta}{dt} \end{aligned} \tag{29}$$

  Si può descrivere sottoforma matriciale nel seguente modo:

$$[A] = \begin{bmatrix} \frac{\partial\varphi_{sd}}{\partial i_{sd}} & \frac{\partial\varphi_{sd}}{\partial i_{sq}} \\ \frac{\partial\varphi_{sq}}{\partial i_{sd}} & \frac{\partial\varphi_{sq}}{\partial i_{sq}} \end{bmatrix} \begin{bmatrix} \frac{di_{sd}}{dt} \\ \frac{di_{sq}}{dt} \end{bmatrix} + \begin{bmatrix} \frac{\partial\varphi_{sd}}{\partial\vartheta} \\ \frac{\partial\varphi_{sq}}{\partial\vartheta} \end{bmatrix} \frac{d\theta}{dt} \tag{30}$$

  A questo punto ricavando le equazioni delle derivate delle correnti dalla *(30)* e inserendola nella *(28)* otteniamo

$$\begin{bmatrix} \frac{di_{sd}}{dt} \\ \frac{di_{sq}}{dt} \end{bmatrix} = [A]^{-1} \left(\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} - R_s \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} - j\omega \begin{bmatrix} \varphi_{sd} \\ \varphi_{sq} \end{bmatrix} - \omega \begin{bmatrix} \frac{\partial\varphi_{sd}}{\partial\vartheta} \\ \frac{\partial\varphi_{sq}}{\partial\vartheta} \end{bmatrix} \right) \tag{31}$$

Quando il modello viene implementato in ambiente digitale (Simulink/Plecs), le equazioni differenziali continue devono essere discretizzate. In questa fase si introduce una semplificazione numerica in cui il modello viene aggiornato a ogni passo di campionamento, perdendo la dipendenza continua dall'angolo rotorico e mantenendola invece nella forma istantanea, valutata punto per punto.

È possibile quindi semplificare ulteriormente l'equazione *(31)* perdendo la dipendenza dalla variabile ϑ.

$$\begin{bmatrix} \frac{di_{sd}}{dt} \\ \frac{di_{sq}}{dt} \end{bmatrix} = [A]^{-1} \left(\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} - R_s \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} - j\omega \begin{bmatrix} \varphi_{sd} \\ \varphi_{sq} \end{bmatrix} \right) \tag{32}$$

- ***Modello flussi inversi***

  Quest'ultimo tipo di modellizzazione è stata considerata soprattutto per verificare che il modello non lineare applicato nelle simulazioni fosse funzionante. Questo modello è un metodo di controllo avanzato che utilizza dati predeterminati, (nel caso delle simulazioni che seguono in tabelle denominate Look-up table o LUT), per determinare le correnti statoriche i_d, i_q richieste in base ai flussi concatenati desiderati (φ_d, φ_q) o alla coppia richiesta. Questo approccio tiene conto delle caratteristiche non lineari del motore, come la saturazione magnetica, che i modelli lineari più semplici spesso trascurano.

In sintesi, i modelli presentati costituiscono la base per la simulazione non lineare del motore. Ognuno offre un diverso compromesso tra accuratezza e complessità computazionale e verranno utilizzati nelle analisi successive, così da assicurare una rappresentazione fedele della macchina sia nelle condizioni stazionarie che no.

2.3 Modulazione PWM

Gli azionamenti implementati in questo elaborato condividono la stessa logica di controllo dell'inverter. In tutte le prove, sia in simulazione sia in configurazione PHIL, la modulazione utilizzata è una PWM a sette intervalli simmetrica, applicata a un inverter trifase a ponte completo. In questo contesto, i tre rami dell'inverter sono comandati a partire dalle tensioni di riferimento del controllore, generate nel sistema di riferimento $d-q$ e successivamente trasformate in tensioni di fase.

La denominazione "PWM unipolare a sette intervalli" non va interpretata nel senso classico della modulazione unipolare dei convertitori monofase (dove la tensione può assumere solo valori positivi o nulli). Nel caso trifase, infatti, ogni fase può assumere combinazioni diverse degli stati dei due interruttori del ramo, e ciò permette la generazione dei due vettori nulli e dei sei vettori attivi tipici dell'inverter trifase a tensione impressa. La variante a sette intervalli si riferisce alla sequenza di commutazione simmetrica adottata all'interno di ciascun periodo di switching, nella quale si impone che:

- ogni ramo commuti una sola volta per periodo,
- la sequenza sia speculare rispetto alla mezzeria del periodo PWM,
- i tempi di applicazione dei due vettori nulli siano identici.

Una possibile sequenza è riportata in Fig. 18.

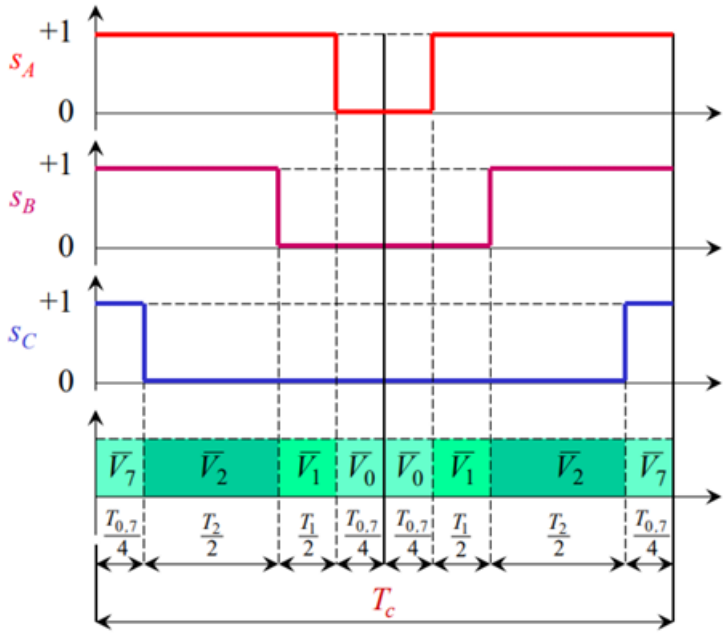


Figura 18:Stati della modulazione PWM unipolare a sette intervalli

Per ottenere questo risultato si deve spostare le modulanti in modo che risultino centrate rispetto all'intervallo [0, 1]. Le modulanti ottenute avranno quindi le seguenti equazioni:

$$\begin{cases} m_A = m_0 + \dfrac{V_{AN,ref}}{V_{dc}} \\ m_B = m_0 + \dfrac{V_{BN,ref}}{V_{dc}} \\ m_C = m_0 + \dfrac{V_{CN,ref}}{V_{dc}} \end{cases} \tag{33}$$

$$m_0 = \frac{1}{2}\left[1 - \min\left(\frac{V_{AN,ref}}{V_{dc}}, \frac{V_{BN,ref}}{V_{dc}}, \frac{V_{CN,ref}}{V_{dc}}\right) - \max\left(\frac{V_{AN,ref}}{V_{dc}}, \frac{V_{BN,ref}}{V_{dc}}, \frac{V_{CN,ref}}{V_{dc}}\right)\right] \tag{34}$$

Dove V_{dc} è la tensione sul bus DC, $V_{AN,BN,CN,ref}$ le tensioni di riferimento derivanti dal controllo e m_0 il termine che trasla le modulanti. Questo metodo riduce la distorsione armonica totale nell'uscita e minimizza le perdite di commutazione migliorando l'efficienza complessiva del sistema. Le modulanti sono confrontate con una portante triangolare di frequenza nota. Il risultato del confronto determina lo stato degli interruttori. In particolare, quando la portante è minore del valore modulante, lo stato dell'interruttore superiore è alto e quello inferiore basso, mentre nella condizione di portante maggiore della modulante, avviene il contrario.

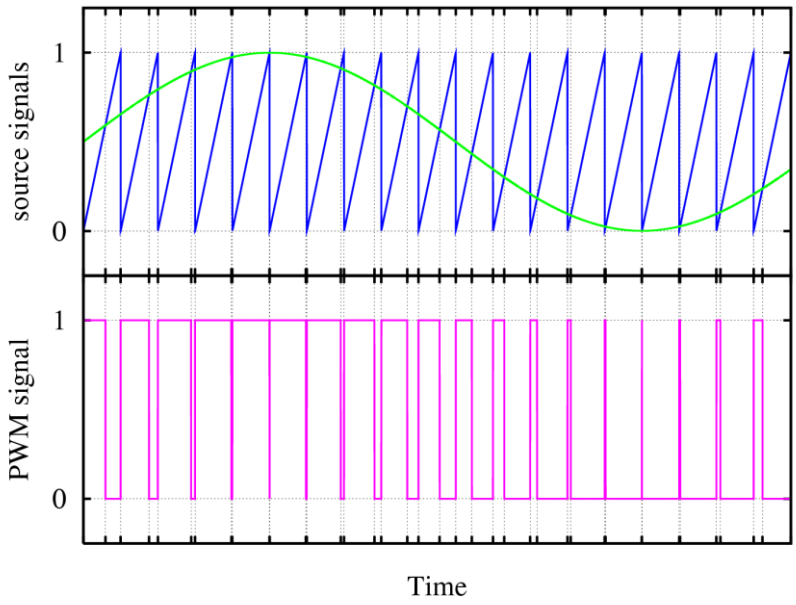


Figura 19: modulazione PWM

Capitolo 3

Simulazioni modelli

Nel capitolo precedente sono stati sviluppati e descritti diversi modelli matematici lineari e non lineari della macchina, ciascuno caratterizzato da un livello differente di complessità e capacità descrittiva. Tali modelli rappresentano la base teorica su cui si fondano le analisi numeriche che verranno ora presentate. L'obiettivo di questo capitolo è valutare il comportamento dinamico ed elettromagnetico della macchina nei vari punti di funzionamento mediante simulazioni che sfruttano i modelli precedentemente introdotti, mettendone in luce differenze, punti di forza e limiti operativi.

Le simulazioni saranno quindi realizzate su vari modelli creati per la verifica della veridicità dei sistemi di controllo in PHIL. Più precisamente i modelli simulati saranno principalmente quattro:

- Modello PLECS PMSM lineare;
- Modello matematico lineare PMSM;
- Modello LUT non lineare;
- Modello flussi inversi non lineare;

Saranno quindi mostrati gli azionamenti implementati sul software PLECS e la realizzazione dei modelli matematici lineari e non lineari; per quest'ultimi sono stati utilizzati software ad elementi finiti utili alla generazione di mappe di flusso che poi sono state implementate sui modelli.

La prima parte di simulazioni riguarda i PMSM lineari con un primo modello preso dalle librerie di PLECS utile poi per la verifica del modello matematico lineare creato. La seconda parte riguarda la costruzione di modelli non lineari (tramite l'utilizzo di diversi software di simulazione e elaborazione dei dati come Matlab e FEMM), in cui si cercherà di verificare la veridicità di essi e risulteranno molto utili per il seguito dell'elaborato

Le simulazioni presentate hanno quindi una duplice finalità: da un lato verificare la correttezza e la coerenza dei modelli matematici sviluppati; dall'altro definire quale struttura modellistica risulti più adatta per gli studi successivi, in termini sia di affidabilità nella rappresentazione del comportamento reale del motore sia di onere computazionale. I risultati che seguiranno costituiranno un passaggio fondamentale per le fasi successive della tesi, nelle quali i modelli verranno impiegati per il progetto e la validazione del sistema di controllo in PHIL.

3.1 Modelli lineari PMSM

In questo sottocapitolo vengono presentate le prime due simulazioni implementate su PLECS dei modelli lineari di motori sincroni a magneti superficiali. Saranno mostrati sia i risultati che la realizzazione dell'intero azionamento (controllo di velocità, inverter, modello motore e sistema meccanico). Entrambe le prove saranno effettuate con gli stessi identici valori di velocità e parametri del motore.

3.1.1 Modello PLECS PMSM a parametri concentrati

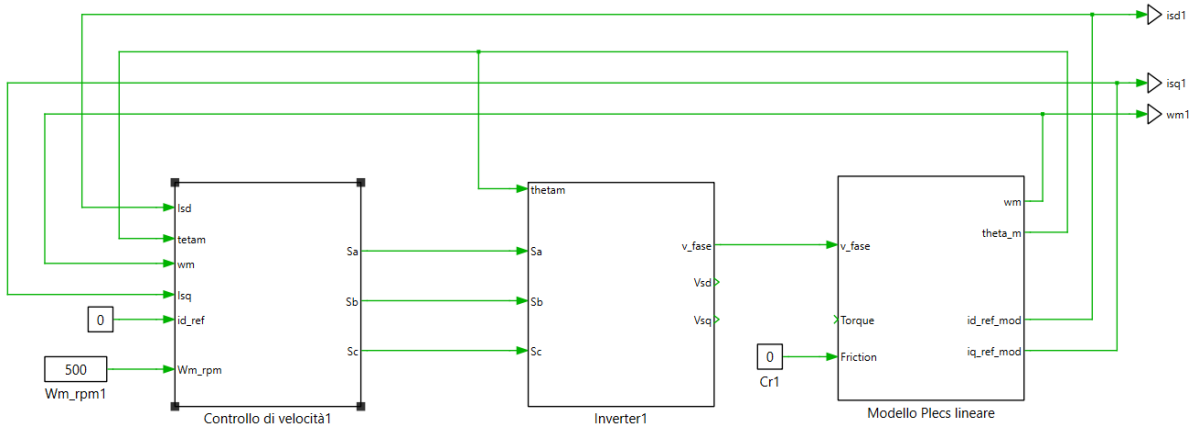


Figura 20: azionamento modello PLECS lineare

Nel sistema rappresentato sono ben visibili i tre sottosistemi che compongono l'azionamento:

- Controllo di velocità;
- Inverter (con modulazione 7 intervalli);
- Modello motore con sistema meccanico.

Prima di mostrare i sottosistemi vengono mostrati i parametri di inizializzazione inseriti

Model initialization commands

```
%%   PARAMETRI MOTORE SINCRONO A MAGNETI SUPERFICIALI

Rs=0.37;
Ls=0.7e-3;
p1=4;

Ki=800;
tau_pi=Ls/Rs;
Kp=tau_pi*Ki;

Cm=0.6;
Jtot=0.4e-3;
b=6.2e-4;
Tsamp=20e-6; %periodo di campionamento
```

Figura 21: parametri di inizializzazione

I valori di K_i e K_p sono stati calcolati in modo da poter tarare i PI del controllo di velocità in maniera corretta.

Controllo di velocità

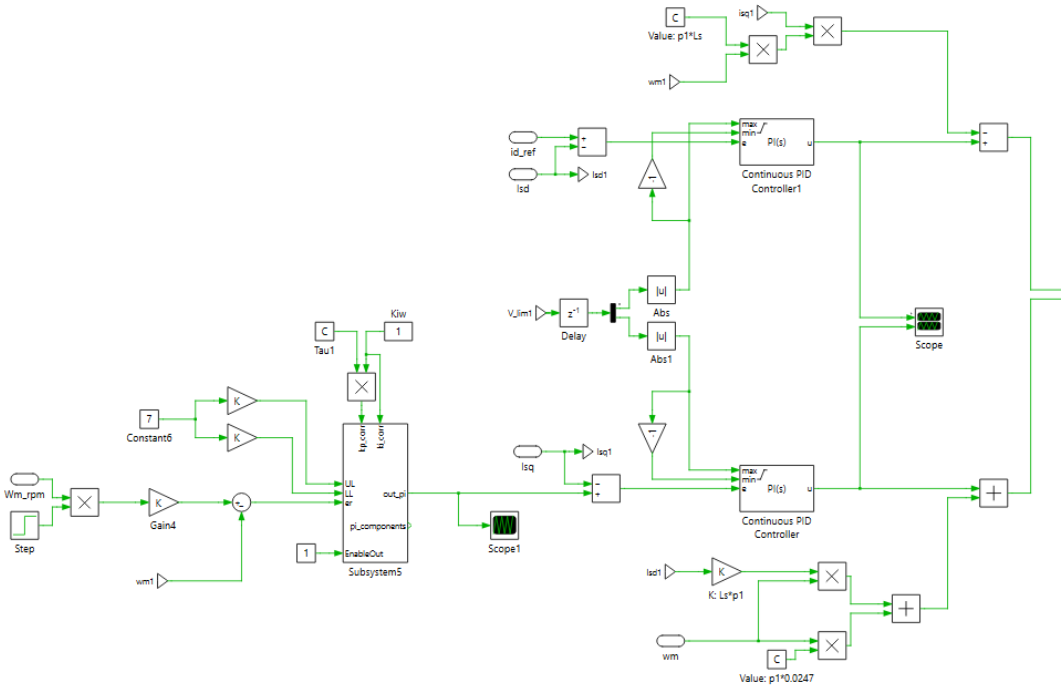


Figura 22: controllo di velocità modello PMSM PLECS

Il controllo di velocità implementato segue quello mostrato in figura 10 e già precedentemente spiegato; in ingresso si ha la velocità in rpm che può essere scelta, vedi figura 22, per poi essere confrontata con quella del rotore (in radianti).

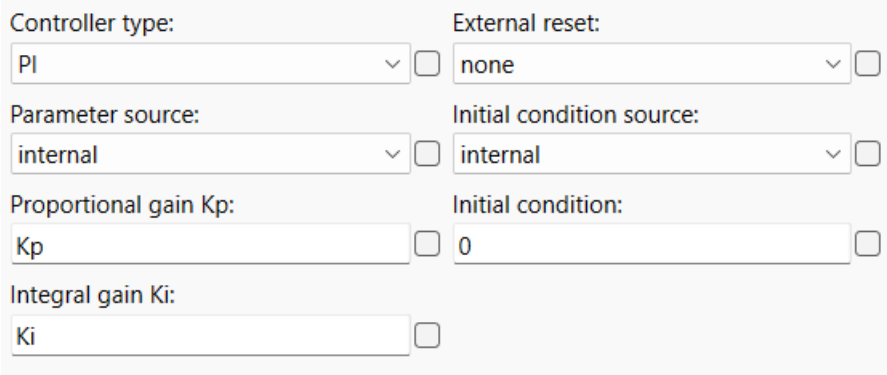


Figura 23: parametri PI

Come descritto in precedenza, i controllori PI in forma discreta sono stati configurati utilizzando i valori di K_p e K_i ricavati dall'analisi preliminare.

Controllo inverter

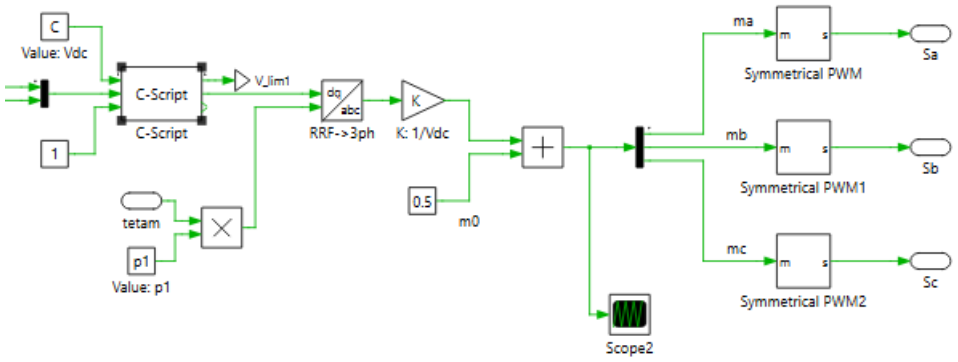


Figura 24: modulazione PWM a 7 intervalli

Il controllo dell'inverter, come spiegato in precedenza, è stato realizzato con la tecnica di modulazione a 7 intervalli impostando poi una modulante $m_0 = 0.5$.

Inverter

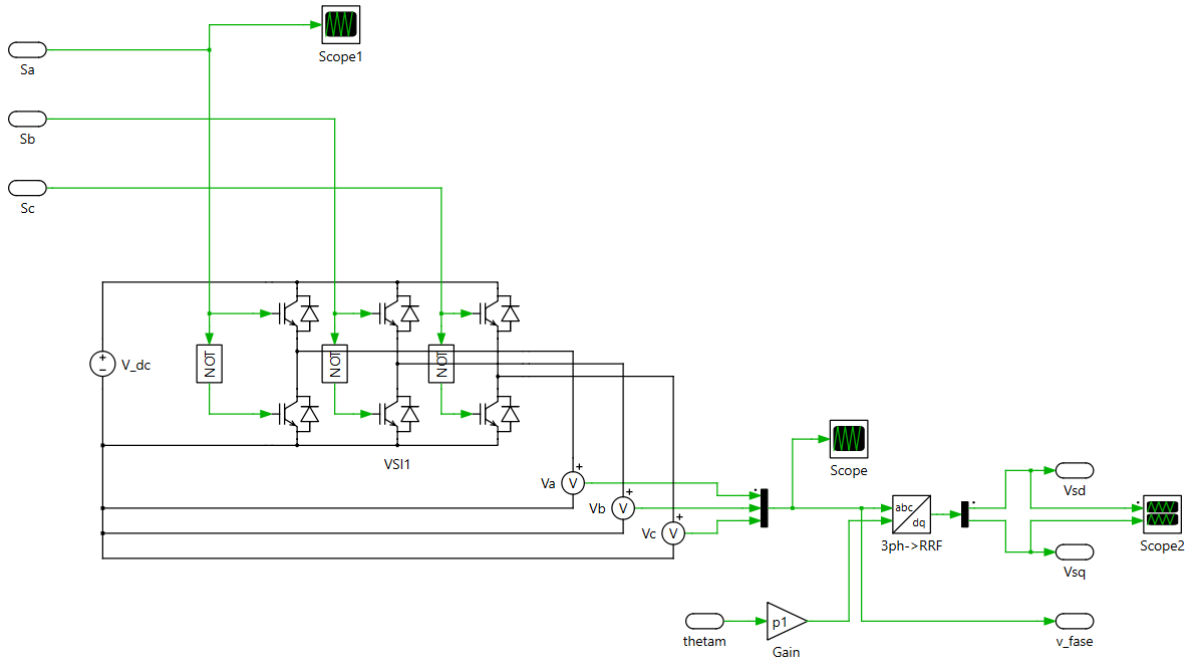


Figura 25: inverter

Le funzioni di commutazione S_a, S_b, S_c vanno a comandare gli interruttori dell'inverter dando in uscita la tensione richiesta che poi viene portato nel riferimento d-q oppure mantenuta con le grandezze di fase in base a quello che il modello del motore richiede.

Modello motore e sistema meccanico

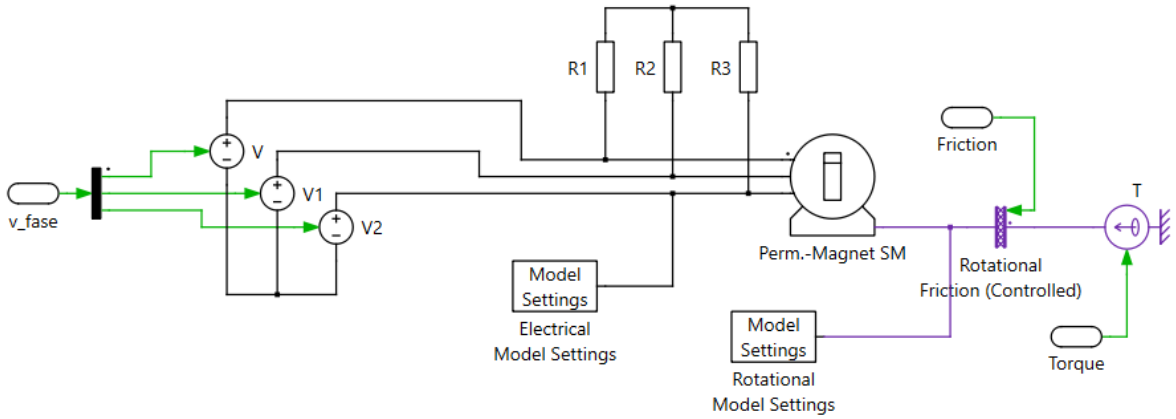


Figura 26: modello motore PMSM e sistema meccanico

Il modello del motore è stato preso dalle librerie di PLECS e collegato alle tensioni di fase in uscita dall'inverter. Tramite i blocchetti "Model Settings" sarà poi possibile studiare le grandezze di uscita di interesse. In aggiunta deve essere collegato un sistema meccanico "Rotational Friction" che simulerà la coppia resistente applicata al rotore.

Permanent-Magnet Synchronous Machine (mask) (link)

Three-phase permanent magnet synchronous machine with sinusoidal back EMF.

Parameters | Assertions

Model: Rotor reference frame	Friction coefficient F (Nms): 6.2e-4
Stator resistance R (Ω): 0.37	Number of pole pairs p: 4
Stator inductance [Ld Lq] (H): [0.7e-3 0.7e-3]	Initial rotor speed wm0 (rad/s): 0
Flux Phi (Vs) induced by magnet: 0.0247	Initial rotor position thm0 (rad): 0
Inertia J (Nms²): 0.4e-3	Initial stator current [isa0 isb0] (A): [0 0]

Figura 27: parametri blocco motore PMSM PLECS

3.1.2 Risultati simulazione modello PLECS

I risultati delle simulazioni che verranno mostrati, saranno fatti per tutti i modelli di motore nelle medesime condizioni:

- Velocità = 750 rpm
- Coppia resistente = 0.2 Nm

Verranno raffigurate rispettivamente per ambe due i modelli:

- tensioni di fase (filtrate con filtro passa basso);
- correnti di fase in riferimenti d-q;
- velocità;
- transitori;

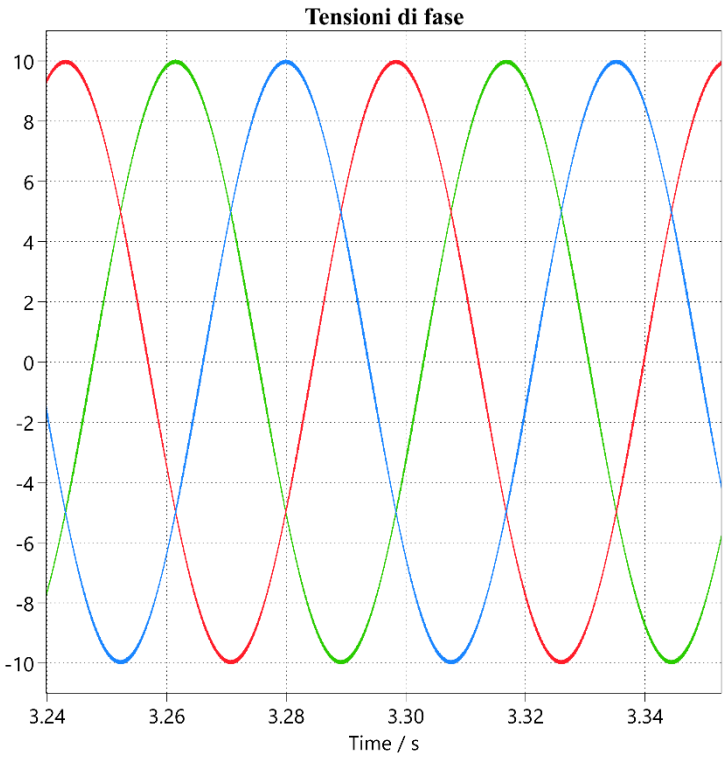


Figura 28: tensioni di fase – blocco motore PMSM PLECS

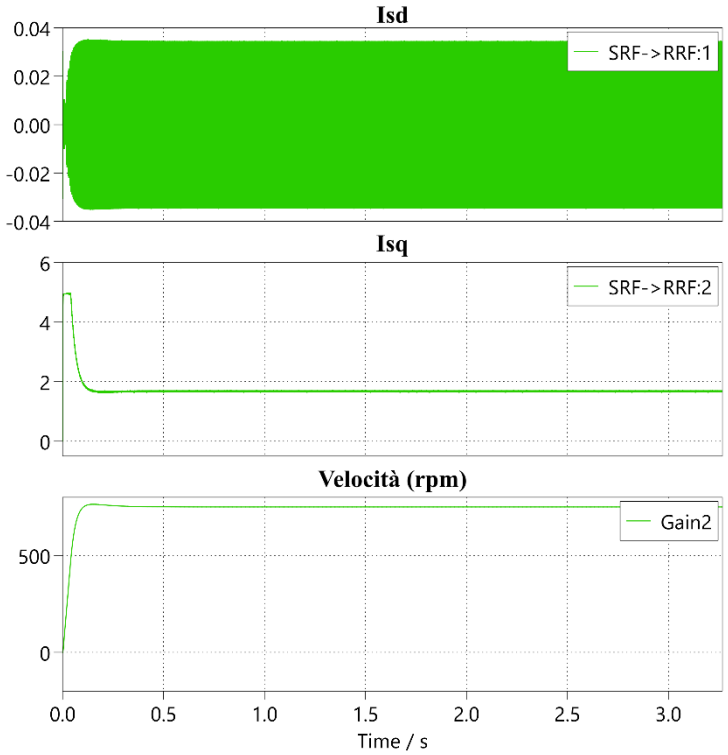


Figura 29: correnti di fase e velocità in simulazione PLECS – blocco motore PMSM PLECS

La corrente i_{sd} non dà alcun contributo di coppia e infatti il suo valore oscilla intorno allo zero, mentre la i_{sq} contribuisce alla creazione di coppia e tende ad aumentare nonostante ci troviamo a vuoto. La velocità è esattamente quella richiesta (750 rpm), avendo quindi la sicurezza che in controllo e le tarature dei PI siano state fatte correttamente. Per osservare meglio l'andamento di correnti e velocità è stato fatto uno zoom andando a modificare la scala temporale:

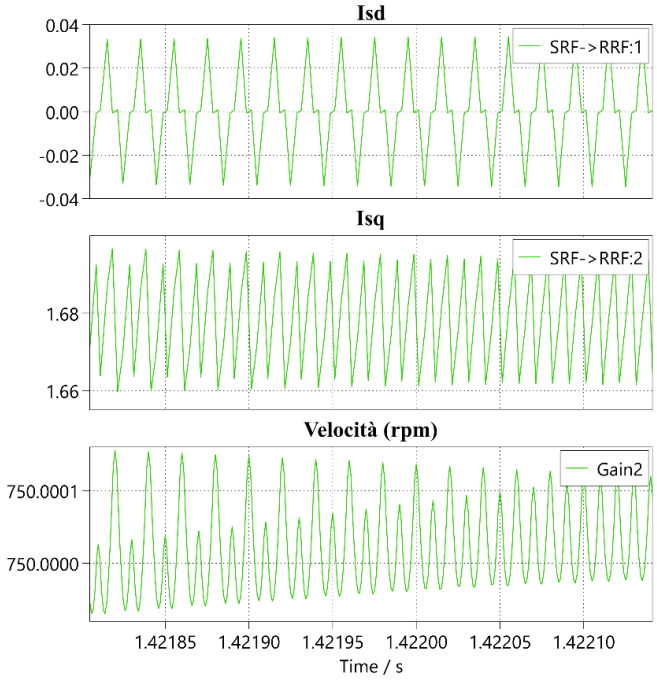


Figura 30: zoom figura 29

Avendo un periodo di campionamento $T_{samp} = 20^{-6}$ l'intervallo di osservazione deve essere molto piccolo per poter riuscire ad apprezzare i reali andamenti delle grandezze.

Un'osservazione viene fatta su quelli che sono i transitori; questi torneranno utili più avanti per il confronto con i modelli simulati con l'architettura PHIL.

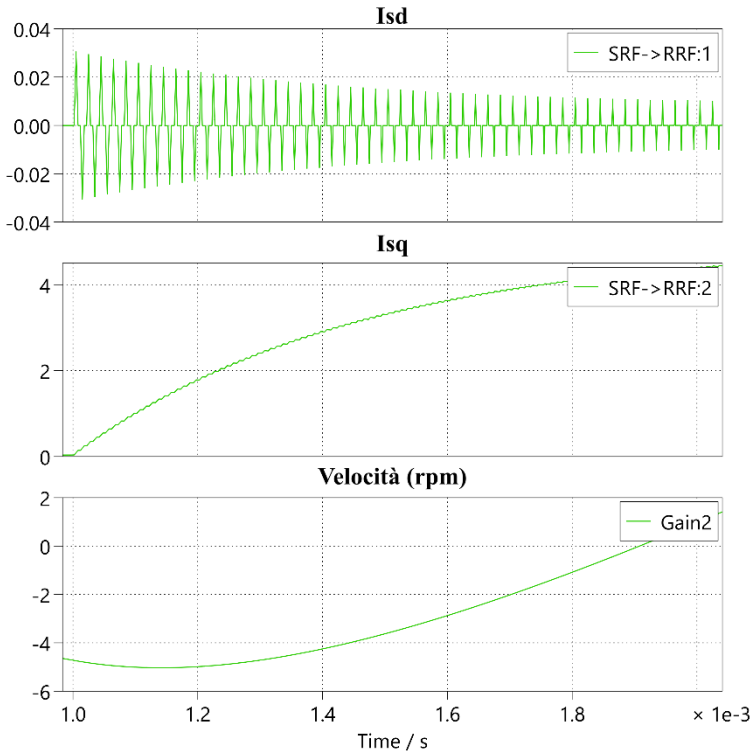


Figura 31: transitori PLECS – blocco motore PMSM PLECS

Si può apprezzare un piccolo spike di corrente che tenderà piano piano a portarsi al valore di corrente corretto e di conseguenza anche una velocità che tenderà a crescere per raggiungere i 750 rpm richiesti. La velocità risulta negativa perché inizialmente deve riuscire a contrastare la coppia resistente imposta di 0.2 Nm.

3.1.3 Modello matematico lineare PMSM

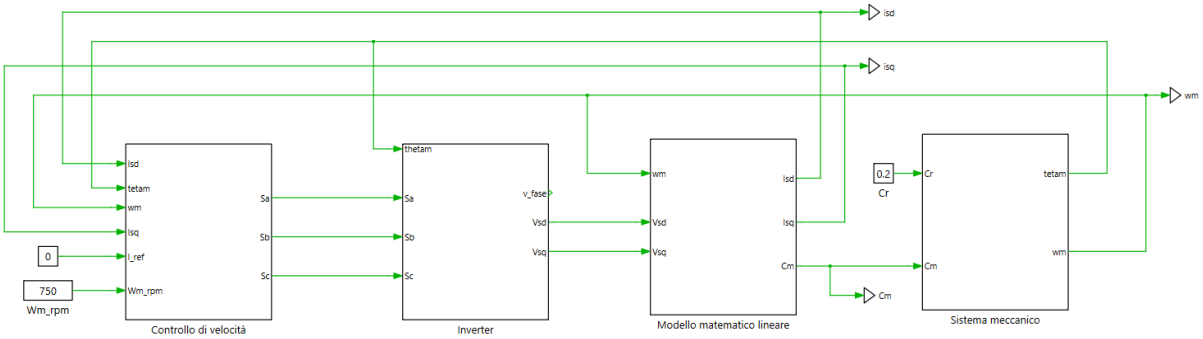


Figura 32: azionamento modello matematico lineare

La struttura dell'azionamento mostrata in figura è sempre la stessa del modello precedente, quello in cui differisce è la presenza di un quarto sottosistema che rappresenta il sistema meccanico. Nel modello precedente il modello del motore comprendeva sia il motore stesso che il sistema meccanico perché il software permette di poter utilizzare entrambi con blocchi già preformati nella libreria.

In questo caso invece il sistema meccanico, proprio come il modello del motore, è stato costruito seguendo i ragionamenti teorici fatti in precedenza. Per comodità vengono sotto riportati solo il modello matematico del motore e il sistema meccanico, in quanto controllo di velocità, controllo dell'inverter e inverter non sono stati modificati.

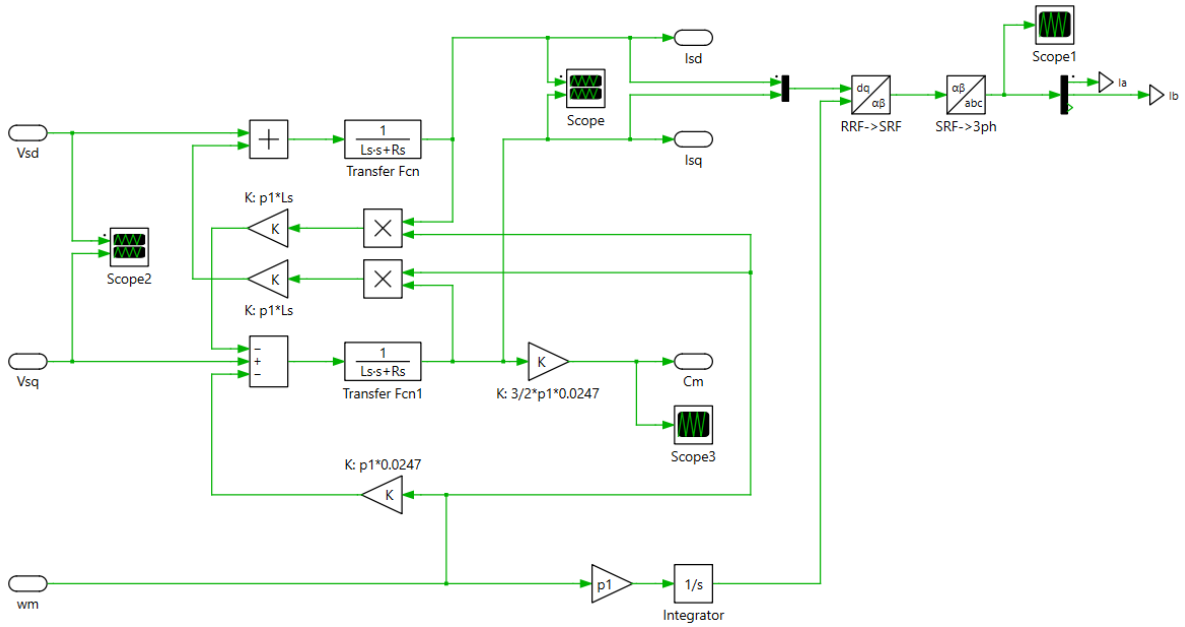


Figura 33: modello matematico motore PMSM implementato su PLECS

Come è possibile notare, la struttura del modello matematico segue esattamente quella che è raffigurata in figura 6 in quanto si tratta esattamente dello stesso tipo di motore.

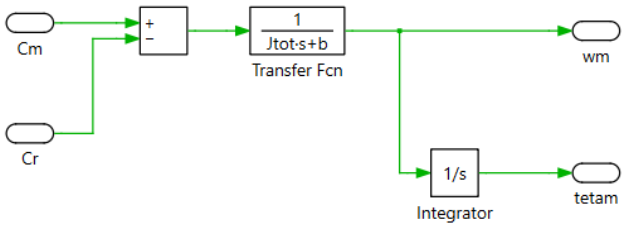


Figura 34: modello sistema meccanico

Il sistema meccanico prende in ingresso la coppia generata dal motore e la funzione di trasferimento che descrive il carico è:

$$G_{load} = \frac{\omega_m}{C_m} = \frac{1}{sJ + b + K_C} \tag{35}$$

Il termine $J = J_{mot} + J_{load}$ è il momento d'inerzia totale del carico, comprensivo di un termine che considera l'inerzia del rotore a vuoto e di uno relativo al carico, supposto essere $J_{load} = 3J_{mot}$. Il coefficiente di attrito, b, è impostato a 10^{-3}. La costante di coppia resistente, K_C, è stata dimensionata per avere, in condizioni di regime del motore, una velocità pari a quella nominale.

$$K_C = \frac{C_m}{\omega_{nom}} - b \tag{36}$$

La funzione *(35)* restituisce la velocità meccanica del rotore; da cui, tramite integrazione della stessa, si ottiene la posizione angolare del rotore θ_m.

3.1.4 Risultati simulazione modello matematico

Come accennato all'inizio del capitolo 3.1.2 le simulazioni sono ripetute nelle medesime condizioni e verranno mostrate le stesse grandezze del modello precedente, con l'aggiunta dell'andamento di coppia che in questo cosa può essere utile per vedere se tutto il sistema è funzionante.

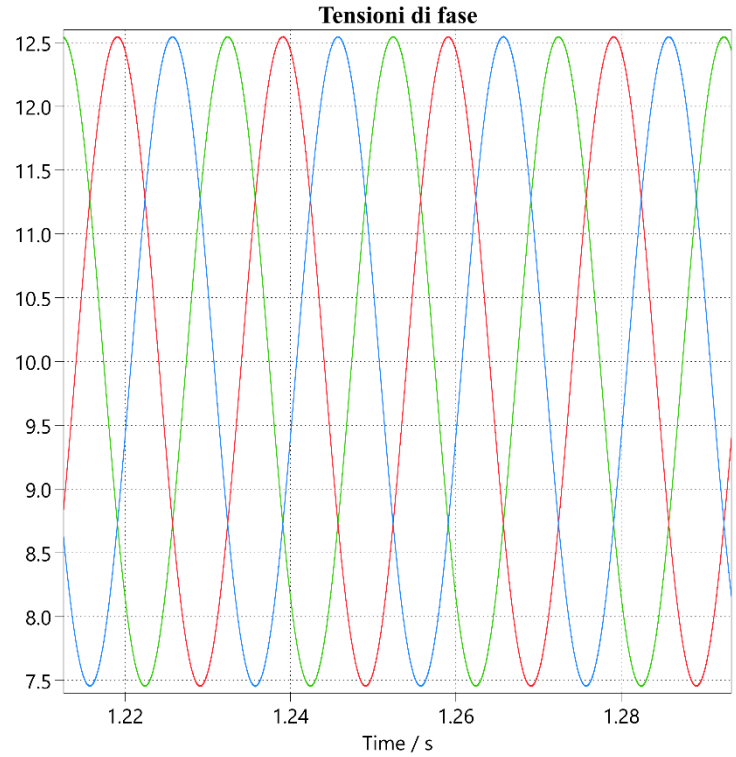


Figura 35: tensioni di fase – modello matematico lineare

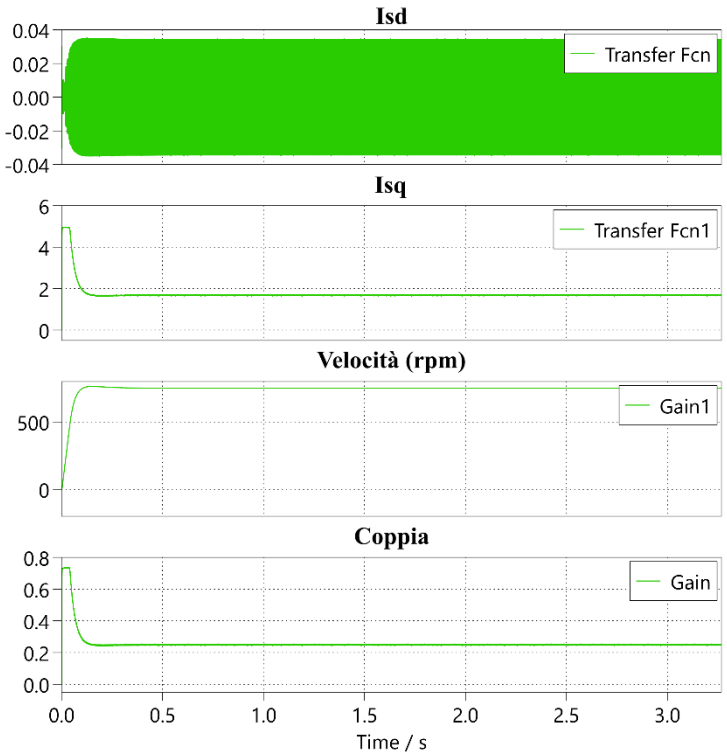


Figura 36: correnti di fase, velocità e coppia in simulazione PLECS – modello matematico lineare PMSM

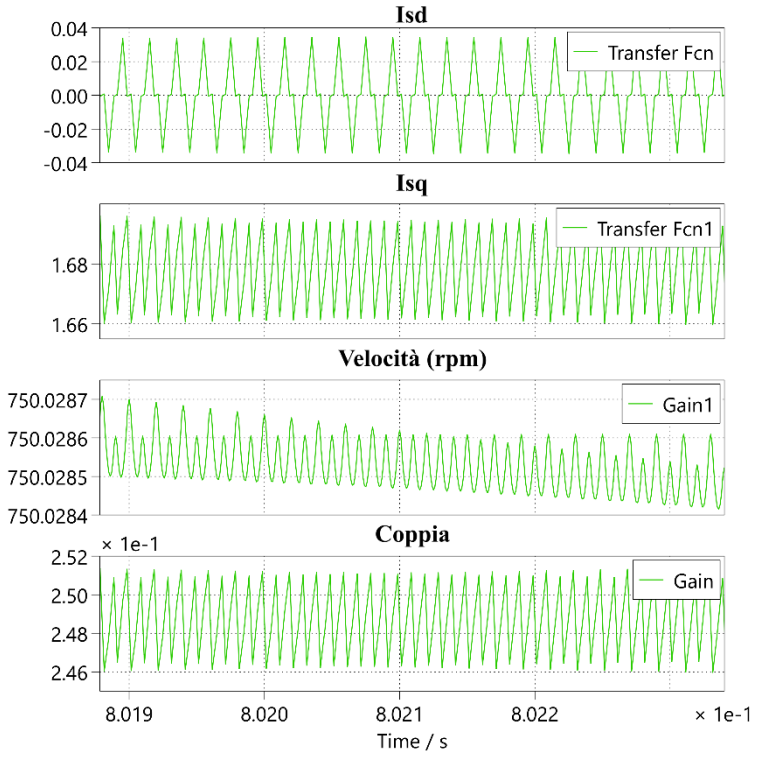


Figura 37: zoom figura 36

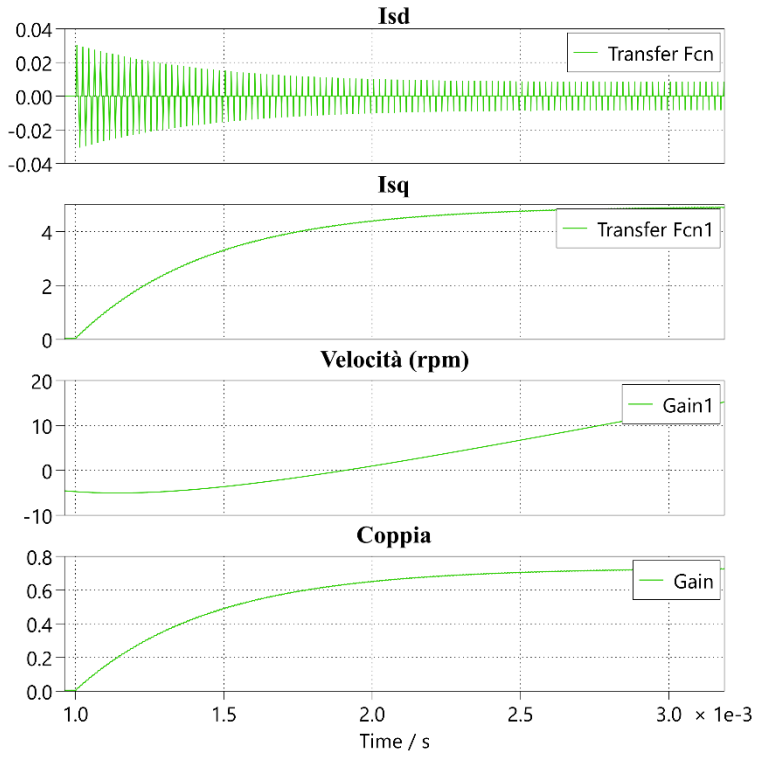


Figura 38: transitori PLECS – modello matematico PMSM

Gli screen dei grafici sono stati presi in momenti temporali diversi rispetto a quelli della simulazione precedente per dimostrare che non fossero gli stessi. Ad ogni modo per facilitare ancora di più il

confronto tra i due modelli e dimostrate che il modello matematico risulta funzionante è stato creato uno scope che mette nello stesso plot le varie grandezze.

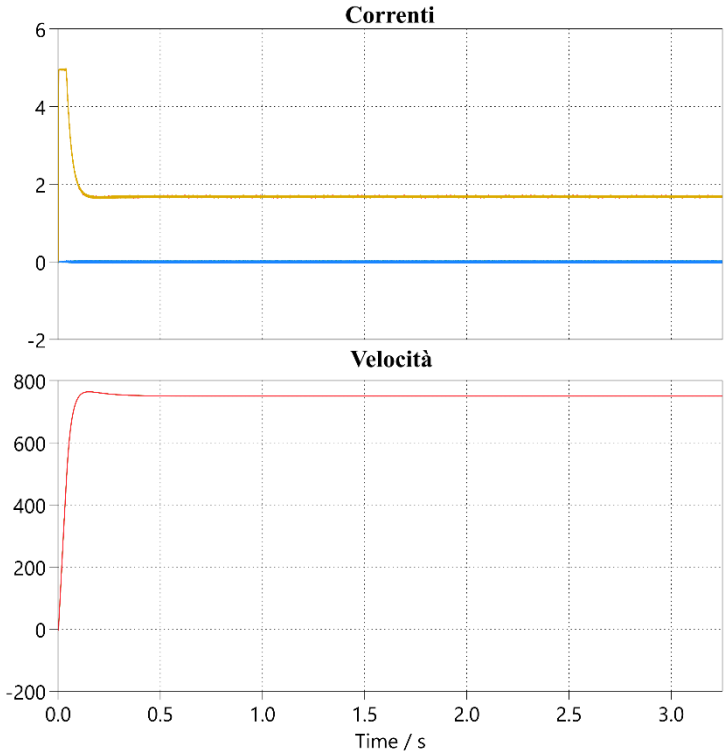


Figura 39: confronto modelli lineari

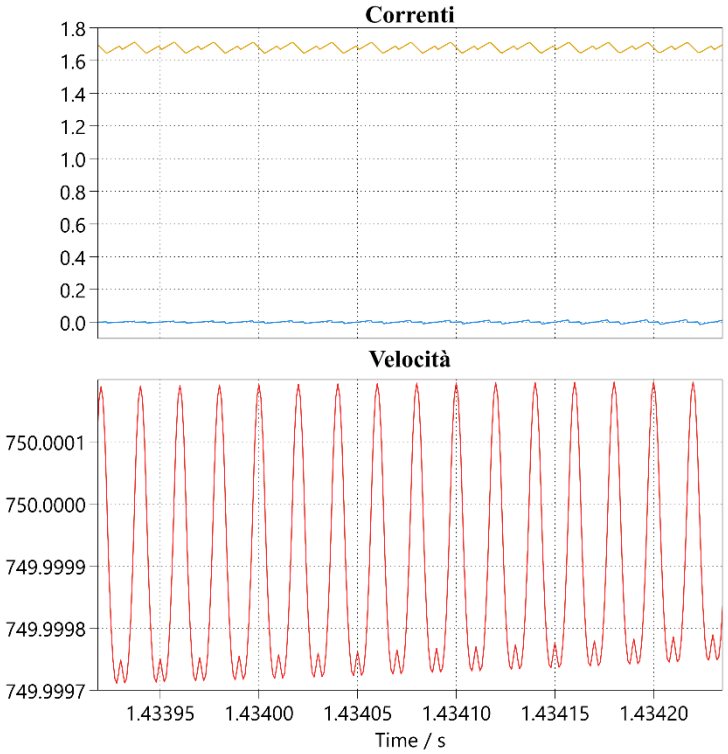


Figura 40: zoom figura 39

Come si può ben vedere dalle figure le correnti e gli andamenti di velocità risultano praticamente sovrapposti al punto di non riuscire a distinguere tutte e quattro le grandezze. Il modello matematico rispetta quindi quello che è il comportamento in regime lineare della macchina e potrà essere utilizzato nelle simulazioni in PHIL.

3.2 Modelli non lineari

Le modellazioni scelte per l'implementazione su PLECS sono state:

- Modello 3D (o modello LUT);
- Modello a flussi inversi;

3.2.1 Modello LUT

Per la costruzione del modello non lineare della macchina a riluttanza pura è stato seguito un approccio basato sull'analisi agli elementi finiti e sulla successiva implementazione della caratteristica magnetica in ambiente di simulazione. Il primo passo ha riguardato la realizzazione del modello FEM bidimensionale tramite il software FEMM, nel quale sono stati definiti la geometria della macchina [3], le caratteristiche dei materiali magnetici e la disposizione degli avvolgimenti. In tale contesto è stato possibile descrivere con precisione il circuito magnetico, tenendo conto delle non linearità dovute alla saturazione, aspetto fondamentale per riprodurre il comportamento reale della macchina a riluttanza.

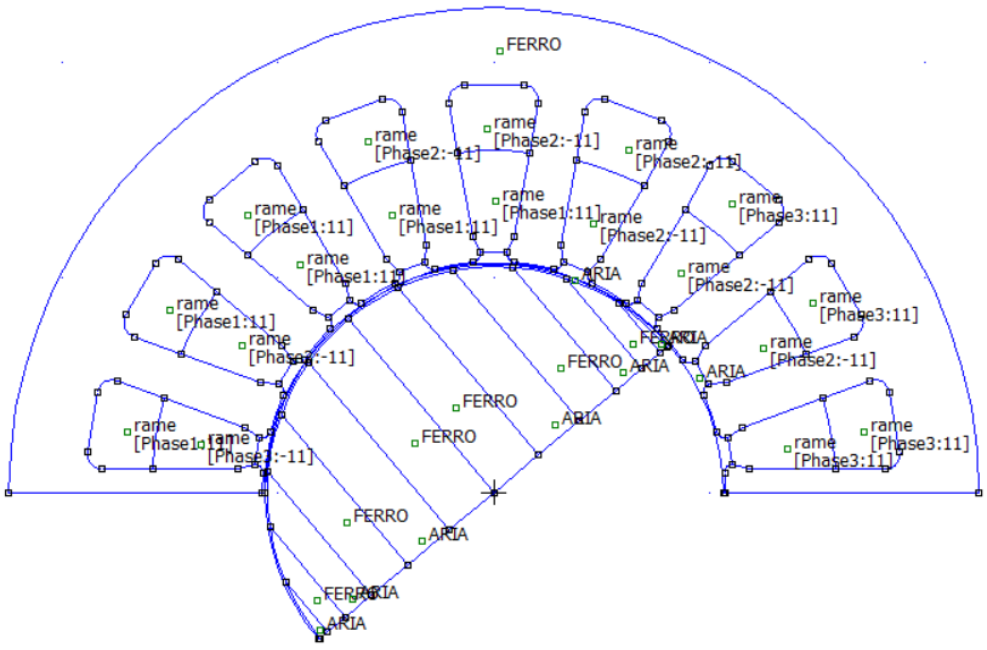


Figura 41: sezione motore Syncrel su FEMM

Una volta costruito il modello FEMM, si è proceduto alla sua integrazione con MATLAB attraverso comandi specifici che consentono di svolgere determinate funzioni direttamente su FEMM e di automatizzare l'intero processo di simulazione. Questa soluzione ha permesso di eseguire in modo sistematico un ampio numero di analisi, variando contemporaneamente la corrente sull'asse diretto, quella sull'asse in quadratura e la posizione angolare del rotore. Sono stati scelti dieci valori per i_{sd}, dieci per i_{sq} e un insieme di posizioni angolari sufficientemente rappresentative, in modo da coprire l'intero spazio operativo del modello (5 punti). Per ciascuna combinazione, MATLAB ha impostato le condizioni di eccitazione nel modello FEMM, fatto eseguire la simulazione e infine estratto i flussi concatenati relativi agli assi d e q.

Qui di seguito vengono riportati gli screen del codice creato per le simulazioni automatiche lanciate da Matlab su FEMM.

```
clear all;
close all;
clc;

p = 1;        % coppie di poli
Nc = 18;

N_punti_teta = 5;     % numero intervalli (default 5)
teta_max_el = 360/6*p;
teta_el_deg  = linspace(0, teta_max_el, N_punti_teta);
teta_el_rad  = teta_el_deg*2*pi/360;
teta_mec_deg = teta_el_deg/p;
teta_mec_iniziale = 0;
teta_el_rad_full=linspace(0,2*pi,6*N_punti_teta);


%modificato-----
Ifase_nominale = 10;
N_punti_idq= 10; %default (10)
id_vector = linspace(-Ifase_nominale, Ifase_nominale , N_punti_idq);
iq_vector = linspace(-Ifase_nominale, Ifase_nominale, N_punti_idq);
%----------------
% open FEMM
openfemm
opendocument('SynRM_assegnato_50mm.FEM');
mi_saveas('temp_completo.fem');
mi_close();
```

Figura 42: realizzazione comando Matlab per simulazione automatica FEMM – parte 1

```
% flussi in DQ (3D)
Flux_d = zeros(N_punti_idq, N_punti_idq, N_punti_teta); % id, iq, theta
Flux_q = zeros(N_punti_idq, N_punti_idq, N_punti_teta); % id, iq, theta
Torque = zeros(N_punti_idq, N_punti_idq, N_punti_teta);

for i_id = 1 : N_punti_idq
    for j_iq = 1 : N_punti_idq
        for k = 1 : N_punti_teta

            opendocument('temp_completo.fem');

            % aggiorna condizioni al contorno
            mi_modifyboundprop('antiperiodic airgap', 10, teta_mec_deg(k)+teta_mec_iniziale);

            % trasformazioni correnti
            i_alfa = id_vector(i_id)*cos(teta_el_rad(k)) - iq_vector(j_iq)*sin(teta_el_rad(k));
            i_beta = id_vector(i_id)*sin(teta_el_rad(k)) + iq_vector(j_iq)*cos(teta_el_rad(k));

            Ifase_A = i_alfa;
            Ifase_B = -0.5*i_alfa + sqrt(3)/2*i_beta;
            Ifase_C = -Ifase_A - Ifase_B;

            mi_modifycircprop('Phase1', 1, Ifase_A);
            mi_modifycircprop('Phase2', 1, Ifase_B);
            mi_modifycircprop('Phase3', 1, Ifase_C);

            mi_saveas('temp_loop.fem');
            mi_analyze(1);
            mi_loadsolution();
```

Figura 43: realizzazione comando Matlab per simulazione automatica FEMM – parte 2

```
            circpropsA = mo_getcircuitproperties('Phase1');
            circpropsB = mo_getcircuitproperties('Phase2');
            circpropsC = mo_getcircuitproperties('Phase3');

            % flussi in abc
            Flux_abc = 2*p*[circpropsA(3), circpropsB(3), circpropsC(3)];

            % alfa-beta
            flux_alfa = 2/3*(Flux_abc(1) - 0.5*Flux_abc(2) - 0.5*Flux_abc(3));
            flux_beta = 2/3*( sqrt(3)/2*Flux_abc(2) - sqrt(3)/2*Flux_abc(3));

            % DQ
            flux_d = flux_alfa * cos(teta_el_rad(k)) + flux_beta * sin(teta_el_rad(k));
            flux_q = -flux_alfa * sin(teta_el_rad(k)) + flux_beta * cos(teta_el_rad(k));

            % salva in 3D (id, iq, theta)
            Flux_d(i_id, j_iq, k) = flux_d;
            Flux_q(i_id, j_iq, k) = flux_q;

            % torque
            Torque(i_id, j_iq, k) = mo_gapintegral('antiperiodic airgap', 0);

            mo_close();
            mi_close();
        end
    end
end
```

Figura 44: realizzazione comando Matlab per simulazione automatica FEMM – parte 3

```
Torque_full=repmat(Torque,[1 1 6]);
Flux_d_full=repmat(Flux_d,[1 1 6]);
Flux_q_full=repmat(Flux_q,[1 1 6]);

%% ============================================================
%   CALCOLO DELLE NON LINEARITÀ (derivate dei flussi)
%   Risultato in 3D (id, iq, theta)
%  ============================================================

% Prealloca matrici induttanze differenziali
Ldd = zeros(N_punti_idq, N_punti_idq, N_punti_teta);
Ldq = zeros(N_punti_idq, N_punti_idq, N_punti_teta);
Lqd = zeros(N_punti_idq, N_punti_idq, N_punti_teta);
Lqq = zeros(N_punti_idq, N_punti_idq, N_punti_teta);

% Derivate numeriche con differenze finite
for i_id = 1:N_punti_idq
    for j_iq = 1:N_punti_idq
        for k = 1:N_punti_teta
            % calcolo derivata in id
            if i_id == 1                    %IN AVANTI
                dflux_d_id = (Flux_d(i_id+1,j_iq,k) - Flux_d(i_id,j_iq,k)) / (id_vector(i_id+1) - id_vector(i_id));
                dflux_q_id = (Flux_q(i_id+1,j_iq,k) - Flux_q(i_id,j_iq,k)) / (id_vector(i_id+1) - id_vector(i_id));
            elseif i_id == N_punti_idq      %INDIETRO
                dflux_d_id = (Flux_d(i_id,j_iq,k) - Flux_d(i_id-1,j_iq,k)) / (id_vector(i_id) - id_vector(i_id-1));
                dflux_q_id = (Flux_q(i_id,j_iq,k) - Flux_q(i_id-1,j_iq,k)) / (id_vector(i_id) - id_vector(i_id-1));
            else                            %CENTRATA
                dflux_d_id = (Flux_d(i_id+1,j_iq,k) - Flux_d(i_id-1,j_iq,k)) / (id_vector(i_id+1) - id_vector(i_id-1));
                dflux_q_id = (Flux_q(i_id+1,j_iq,k) - Flux_q(i_id-1,j_iq,k)) / (id_vector(i_id+1) - id_vector(i_id-1));
```

Figura 45: realizzazione comando Matlab per simulazione automatica FEMM – parte 4

```
            end

            % calcolo derivata in iq, UGUALE id
            if j_iq == 1
                dflux_d_iq = (Flux_d(i_id,j_iq+1,k) - Flux_d(i_id,j_iq,k)) / (iq_vector(j_iq+1) - iq_vector(j_iq));
                dflux_q_iq = (Flux_q(i_id,j_iq+1,k) - Flux_q(i_id,j_iq,k)) / (iq_vector(j_iq+1) - iq_vector(j_iq));
            elseif j_iq == N_punti_idq
                dflux_d_iq = (Flux_d(i_id,j_iq,k) - Flux_d(i_id,j_iq-1,k)) / (iq_vector(j_iq) - iq_vector(j_iq-1));
                dflux_q_iq = (Flux_q(i_id,j_iq,k) - Flux_q(i_id,j_iq-1,k)) / (iq_vector(j_iq) - iq_vector(j_iq-1));
            else
                dflux_d_iq = (Flux_d(i_id,j_iq+1,k) - Flux_d(i_id,j_iq-1,k)) / (iq_vector(j_iq+1) - iq_vector(j_iq-1));
                dflux_q_iq = (Flux_q(i_id,j_iq+1,k) - Flux_q(i_id,j_iq-1,k)) / (iq_vector(j_iq+1) - iq_vector(j_iq-1));
            end

            % salva induttanze differenziali (3D)
            Ldd(i_id,j_iq,k) = dflux_d_id;
            Ldq(i_id,j_iq,k) = dflux_d_iq;
            Lqd(i_id,j_iq,k) = dflux_q_id;
            Lqq(i_id,j_iq,k) = dflux_q_iq;
        end
    end
end

Ldd_full=repmat(Ldd,[1 1 6]);
Lqq_full=repmat(Lqq,[1 1 6]);
Ldq_full=repmat(Ldq,[1 1 6]);
Lqd_full=repmat(Lqd,[1 1 6]);
```

Figura 46: realizzazione comando Matlab per simulazione automatica FEMM – parte 5

```
142  save('Dati_LUT_finale.mat','Ldd_full','Lqq_full','Ldq_full','Lqd_full','Flux_q_full','Flux_d_full','id_vector','iq_vecto
143
144  %   PLOT DEI FLUSSI λd E λq
145  k = 2;
146  % estrai slice dei flussi per θ scelto
147  Flux_d_slice = Flux_d(:,:,k);
148  Flux_q_slice = Flux_q(:,:,k);
149
150  % Plot 3D λd
151  figure;
152  surf(id_vector, iq_vector, Flux_d_slice');
153  xlabel('i_d [A]');
154  ylabel('i_q [A]');
155  zlabel('\lambda_d [Wb]');
156  title(['Flusso \lambda_d per \theta = ' num2str(teta_el_deg(k)) '°']);
157  grid on;
158  shading interp;
159  colorbar;
160
161  % Plot 3D λq
162  figure;
163  surf(id_vector, iq_vector, Flux_q_slice');
164  xlabel('i_d [A]');
165  ylabel('i_q [A]');
166  zlabel('\lambda_q [Wb]');
167  title(['Flusso \lambda_q per \theta = ' num2str(teta_el_deg(k)) '°']);
168  grid on;
169  shading interp;
170  colorbar;
```

Figura 47: realizzazione comando Matlab per simulazione automatica FEMM – parte 6

L'insieme dei dati ottenuti dalle simulazioni è stato poi riorganizzato in forma di mappe tridimensionali di flusso, espresse come funzioni di i_{sd}, i_{sq} e della posizione rotorica ϑ. Queste superfici descrivono in modo completo la dipendenza non lineare della macchina dalle variabili correnti e meccaniche, fornendo così una rappresentazione accurata del comportamento magnetico anche in condizioni di saturazione. L'interpolazione dei punti simulati permette inoltre di ricavare il valore del flusso per qualunque combinazione delle variabili all'interno del dominio analizzato.

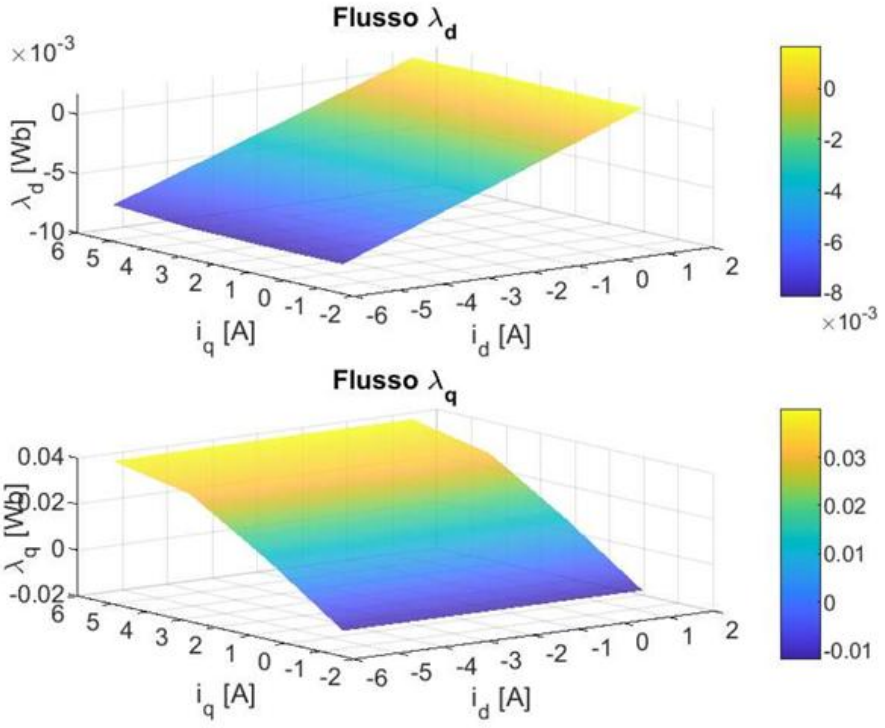


Figura 48: andamento flussi d e q ottenuto con derivate parziali

Sono state calcolate e graficate per ogni asse (d e q) e per ogni punto della mappa, le derivate parziali come:

- Differenza in avanti sul primo punto della griglia
- Differenza all'indietro sull'ultimo punto
- Differenza centrata nei punti interni

Esempio per $\frac{\partial \varphi_d}{\partial i_d}$:

$$\frac{\partial \varphi_d(i_d,j)}{\partial i_d} = \begin{cases} \dfrac{\varphi_d(i_d+1,j)-\varphi_d(i_d,j)}{i_d(i+1)-i_d(i)} & \text{(differenza in avanti)} \\ \dfrac{\varphi_d(i_d,j)-\varphi_d(i_d-1,j)}{i_d(i)-i_d(i-1)} & \text{(differenza indietro)} \\ \dfrac{\varphi_d(i_d+1,j)-\varphi_d(i_d-1,j)}{i_d(i+1)-i_d(i-1)} & \text{(differenza centrata)} \end{cases}$$

- Lo stesso procedimento è stato applicato per $\frac{\partial \varphi_d}{\partial i_q}$, $\frac{\partial \varphi_q}{\partial i_d}$, $\frac{\partial \varphi_q}{\partial i_q}$ ottenendo così le quattro matrici tridimensionali $L_{dd}, L_{dq}, L_{qd}, L_{qq}$:

$$L_{dd}(i_d,i_q,\theta), L_{dq}(i_d,i_q,\theta), L_{qd}(i_d,i_q,\theta), L_{qq}(i_d,i_q,\theta)$$

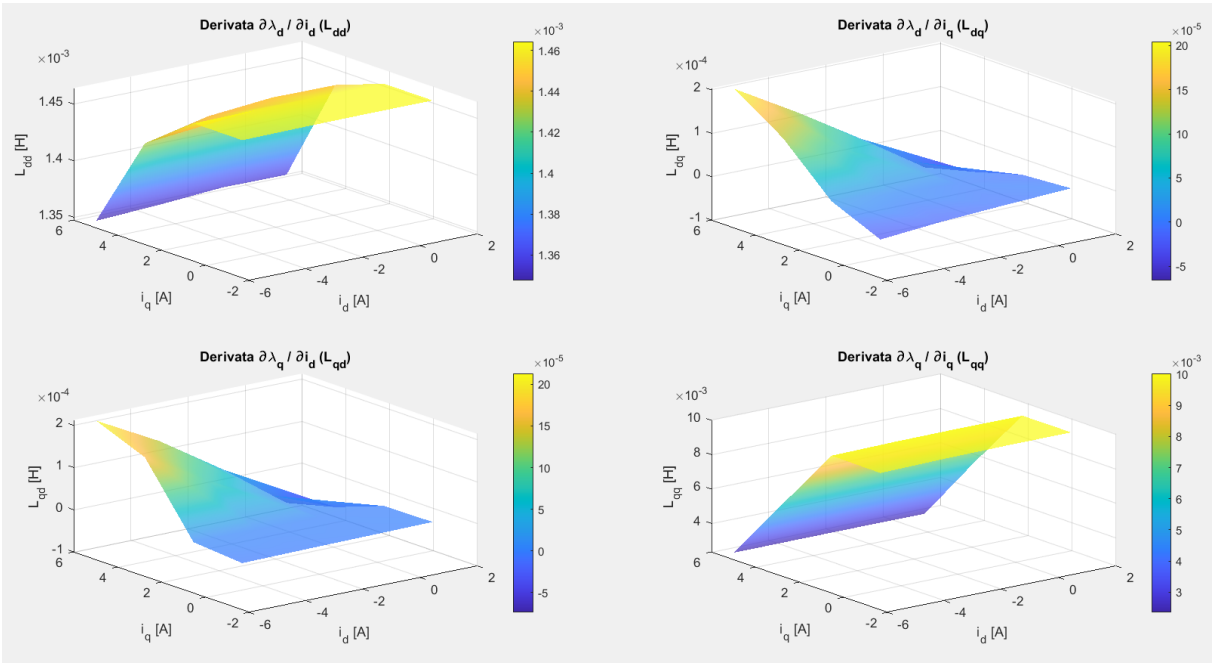


Figura 49: andamenti derivate parziali calcolate

Le mappe di flusso così ottenute sono state infine importate in PLECS, dove è stato costruito il modello non lineare della macchina. L'utilizzo delle look-up table permette al simulatore di determinare istante per istante i valori di φ_{sd} *e* φ_{sq} in funzione delle correnti e della posizione rotorica, garantendo che il comportamento del modello rifletta fedelmente quello ricavato dalle simulazioni FEM. Da tali

grandezze vengono poi calcolate le relazioni elettromagnetiche della macchina, inclusa la coppia, che risulta quindi coerente con la natura fortemente non lineare della macchina reale.

3.2.2 Implementazione su PLECS modello LUT

Nel lato pratico per implementare il tutto su Plecs, come prima cosa bisogna definire nei parametri di simulazione il file .mat generato su Matlab (vedi figura 47 comando save), in cui vengono salvati tutti i vettori e matrici di nostro interesse e i nuovi valori del motore.

Workspace

Name	Value
Flux_d_medio	10x10 double
Flux_q_medio	10x10 double
id_vector	[-10,-7.7778,-5.5...
iq_vector	[-10,-7.7778,-5.5...
Ldd	10x10 double
Ldq	10x10 double
Lqd	10x10 double
Lqq	10x10 double
Torque_fluxes	10x10 double
Torque_medio	10x10 double

Figura 50: contenuto file .mat

Utilizzando il comando load sarà possibile importare il file su Plecs (vedi figura seguente)

Model initialization commands

```
1  load('Dati_LUT_finale_medio.mat');
2
3  %%   PARAMETRI MOTORE A RILUTTANZA   %%
4
5  Ld=2.9e-3;%H
6  Lq=8.4e-3;
7  R=1.35;%ohm
8
9  tau_d=Ld/R;
10 tau_q=Lq/R;
11 Ki_d=800;
12 Ki_q=Ki_d;
13
14 Kp_d=Ki_d*tau_d;
15 Kp_q=Ki_q*tau_q;
16
17 p2=1;
```

Figura 51: simulation parameters SyncRel

A questo punto è possibile realizzare l'intero azionamento proprio come nel modello lineare tenendo però conto che il controllo di velocità cambierà perché non stiamo più simulando la PMSM.

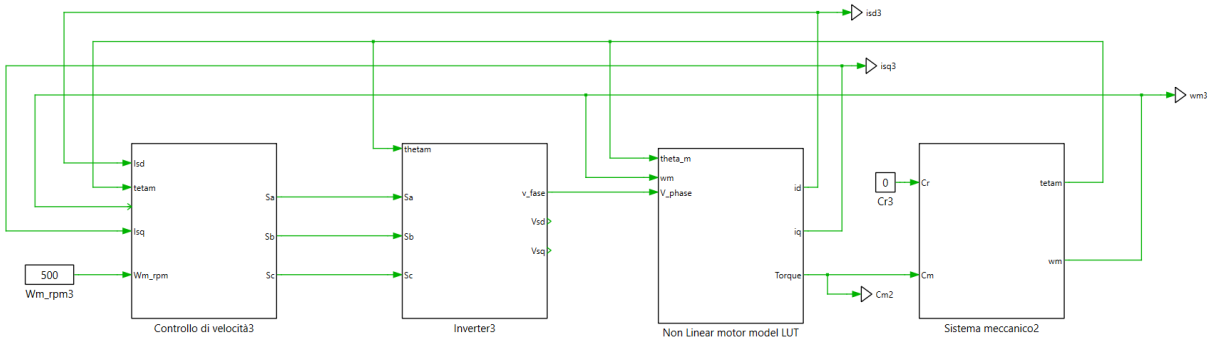


Figura 52: azionamento modello LUT

Controllo velocità

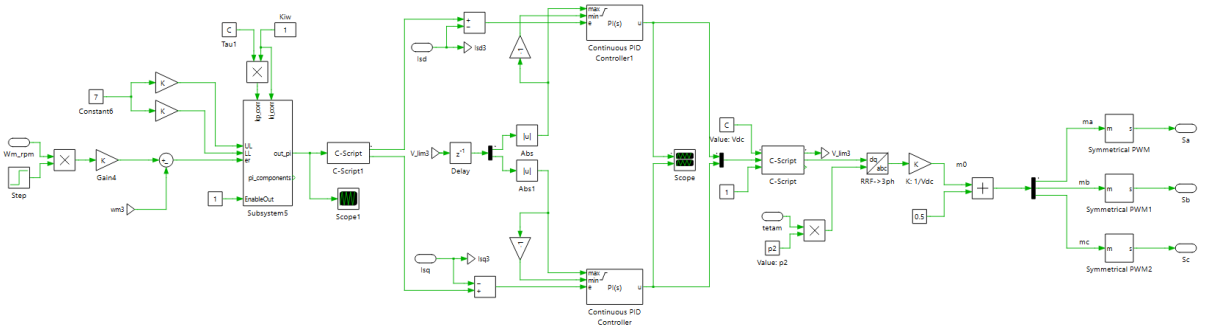


Figura 53: controllo di velocità modello LUT

Il controllo di velocità implementato, come detto, risulta differente rispetto ai modelli precedenti, questo perché il controllo di velocità di una macchina sincrona dipende in modo diretto dal principio fisico con cui essa genera la coppia elettromagnetica. Nel caso di una macchina sincrona a magneti permanenti con magneti superficiali (PMSM), il flusso rotorico è costante e indipendente dalla corrente, poiché generato dai magneti stessi. La coppia si ottiene quindi esclusivamente attraverso la corrente in quadratura i_q, secondo la relazione:

$$C_m = \frac{3}{2} p \varphi_m i_q \tag{37}$$

dove φ_m è il flusso prodotto dai magneti.

Ne consegue che la componente diretta della corrente i_d viene mantenuta prossima a zero (tranne nelle regioni di indebolimento di campo), e il controllo di velocità agisce unicamente sul riferimento di i_q. In altre parole, il regolatore di velocità genera un riferimento di coppia che viene tradotto direttamente in un valore di corrente in quadratura proporzionale alla coppia desiderata.

Diversa è la situazione nella macchina sincrona a riluttanza pura (SyncRel), dove il rotore non presenta magneti e la produzione di coppia avviene esclusivamente per effetto della variazione di riluttanza magnetica tra l'asse diretto e quello in quadratura. In questo caso la coppia è data da:

$$C_m = \frac{3}{2} p (L_d - L_q) i_d i_q \tag{38}$$

La presenza simultanea delle due correnti i_d e i_q rende il controllo più complesso: entrambe concorrono alla generazione di coppia e il loro rapporto influenza direttamente il rendimento e la stabilità del sistema. Inoltre, la macchina a riluttanza presenta un comportamento fortemente non lineare, in quanto le induttanze L_d e L_q variano con la corrente per effetto della saturazione magnetica.

Per queste ragioni, il controllo di velocità di una macchina a riluttanza pura deve essere progettato tenendo conto di tali non linearità e della necessità di ottimizzare la combinazione delle correnti. Nella pratica, il regolatore di velocità fornisce ancora un riferimento di coppia, ma quest'ultima viene poi tradotta in un punto di lavoro nel piano delle correnti secondo una strategia di *Maximum Torque Per Ampere* (MTPA). Tale strategia consente di massimizzare la coppia per un determinato valore di corrente, migliorando l'efficienza del sistema e compensando la mancanza di un flusso magnetico permanente.

In sintesi, mentre nella PMSM il controllo di velocità si riduce alla regolazione della corrente in quadratura, nella macchina a riluttanza pura è necessario un controllo vettoriale a due assi coordinati, nel quale la gestione congiunta di i_d e i_q diventa essenziale per garantire le prestazioni dinamiche e l'efficienza desiderate. Questa differenza sostanziale rende il controllo della SyncRel più articolato, ma al tempo stesso più flessibile e adatto a macchine prive di magneti permanenti, dove la generazione di coppia è interamente affidata al fenomeno di riluttanza variabile.

Modello LUT Riluttanza pura

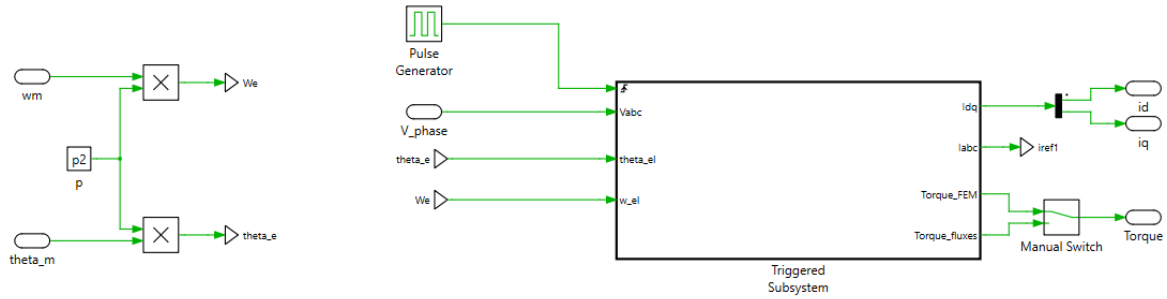


Figura 54: triggered subsystem modello LUT

Il modello è stato racchiuso all'interno di un sottosistema triggerato che ad ogni impulso procede con la risoluzione dello script implementato per la risoluzione. Il trigger è comandato da un'onda quadra con periodo 20 μs, il codice dentro al blocco viene eseguito ad ogni impulso positivo e quindi ogni 20 μs.

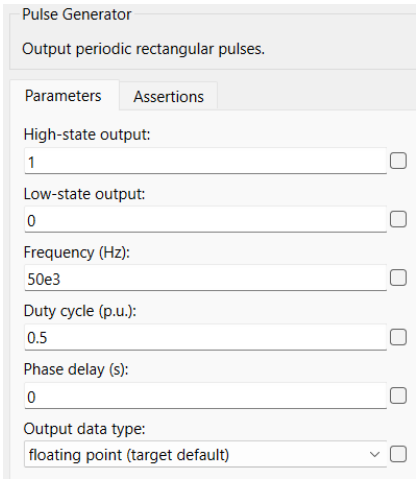


Figura 55: parametri impulse generator

Ora è possibile visualizzare il modello LUT costruito con particolari tabelle in cui è possibile inserire le grandezze vettoriali calcolate nel file .mat, che prendono il nome di “Look-up table” o “LUT” (da cui prende il nome il modello implementato).

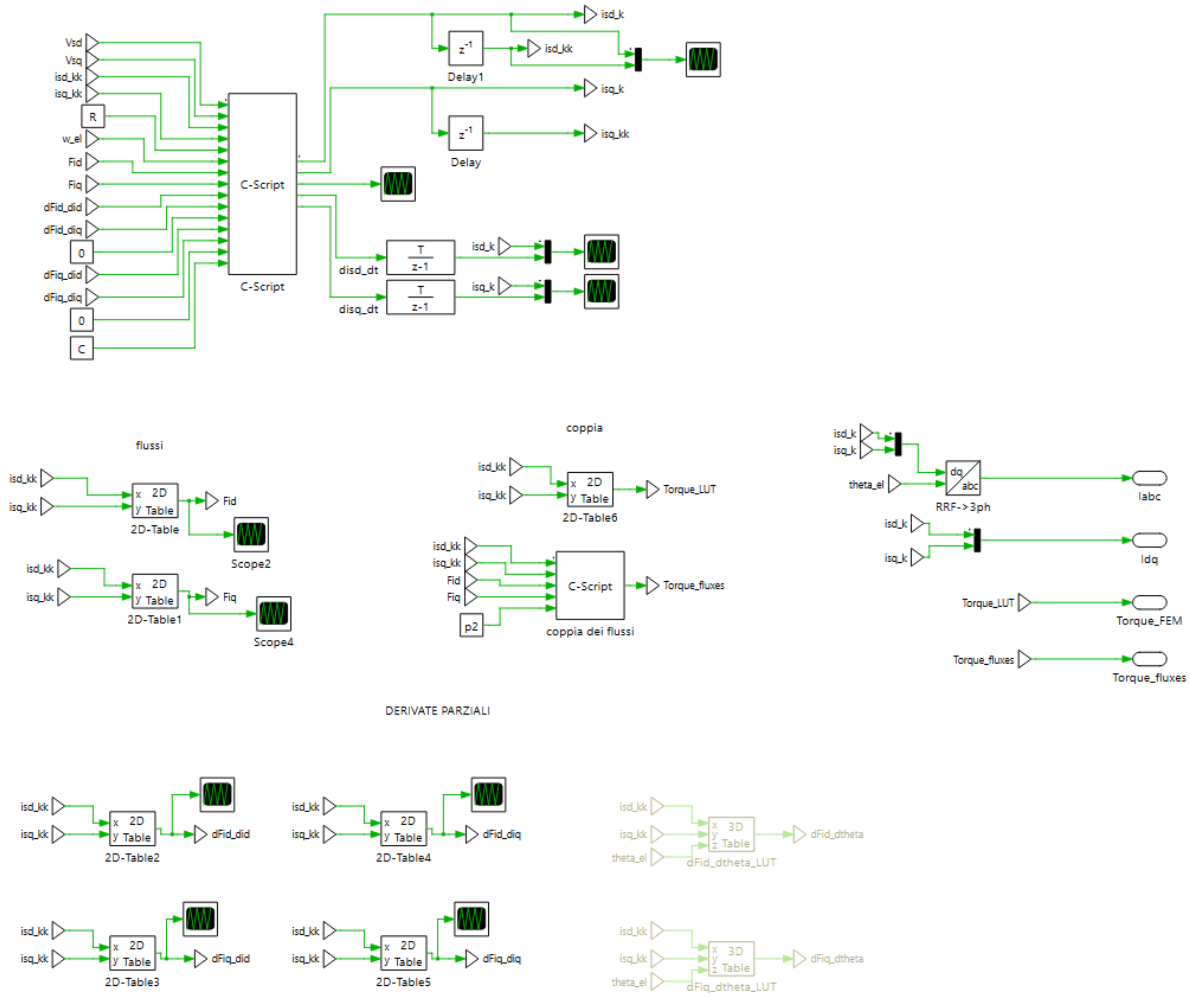


Figura 56: modello LUT

Il modello è stato strutturato in modo che ad ogni impulso i blocchi c-script diano in uscita i risultati richiesti: fondamentale è il calcolo delle correnti e delle derivate parziali delle correnti che permettono di aggiornare i valori in uscita dalle lut.

Setup Code

Output function code

```
double A[2][2]={{dFid_did, dFid_diq},{dFiq_did, dFiq_diq}};

error=0;
//inverto la matrice A
   //calcolo in determinante
   double D = A[0][0]*A[1][1] - A[0][1]*A[1][0];
   //int i=0;
   //int j=0;

   //calcolo l'inversa se possible
   if (D != 0){
      invA[0][0]= A[1][1]/D;
      invA[0][1]= - A[0][1]/D;
      invA[1][0]= - A[1][0]/D;
      invA[1][1]= A[0][0]/D;

/*    for (i=0; i<2; i++){
         for(j=0; j<2 ; j++){
            invA[i][j]=invA[i][j]/D;
         }
      }  */
   }
   else{
   error=1;
   Isd_k=0;
   Isq_k=0;
   }

//definisco vettori per il calcolo
   double Vs[2]={Vsd,Vsq};
   double Is_kk[2]={Isd_kk,Isq_kk};
   double Fis[2]={-Fiq, Fid};
   double dFis_dtheta[2]={dFid_dtheta,dFiq_dtheta};

//calcolo: is_k= is_kk + invA*(vs - Rs Is_kk - wel Fis - wel *dFis_dteta)*Tc
   a = (Vs[0]-Rs*Is_kk[0]-w_el*Fis[0]-w_el*dFis_dtheta[0])*Tc;
   b = (Vs[1]-Rs*Is_kk[1]-w_el*Fis[1]-w_el*dFis_dtheta[1])*Tc;
   Isd_k = Is_kk[0] + invA[0][0]*a+invA[0][1]*b;
   Isq_k = Is_kk[1] + invA[1][0]*a+invA[1][1]*b;
   disd_dt=(invA[0][0]*a+invA[0][1]*b)/Tc;
   disq_dt=(invA[1][0]*a+invA[1][1]*b)/Tc;

OutputSignal(0,0)=Isd_k;
OutputSignal(1,0)=Isq_k;
OutputSignal(2,0)=error;
OutputSignal(3,0)=disd_dt;
OutputSignal(4,0)=disq_dt;
```

Figura 57: c-script modello LUT per calcolo correnti e derivate parziali

Per calcolare i valori delle correnti si deve per prima cosa verificare che la matrice [A] sia invertibile; se lo è allora si possono implementare i passaggi teorici spiegati precedentemente in cui si calcoleranno i valori di correnti e derivata di corrente utili al funzionamento delle LUT.

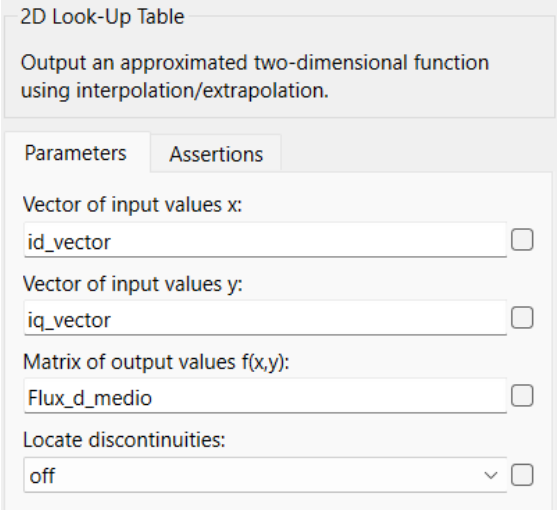


Figura 58: inserimento parametri nelle LUT – calcolo φ_d

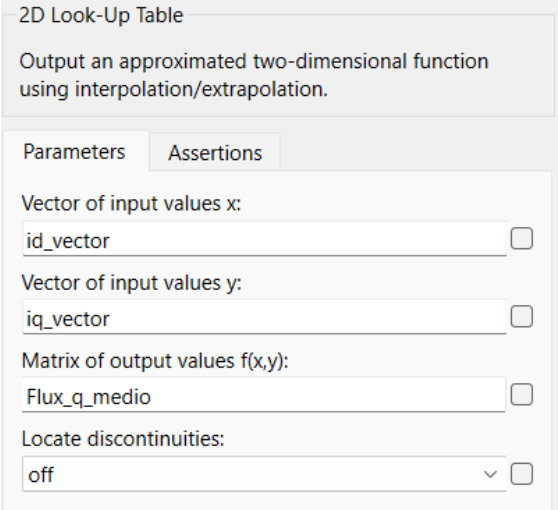


Figura 59: inserimento parametri nelle LUT – calcolo φ_q

In linea generale, il principio di funzionamento delle LUT memorizza una serie di valori discreti che rappresentano la grandezza di uscita in funzione di una o più variabili di ingresso. Durante la simulazione, PLECS riceve i valori correnti delle variabili di ingresso, individua i punti della tabella più vicini e calcola il valore di uscita corrispondente tramite interpolazione (lineare o bilineare, a seconda della dimensione della tabella). Nelle figure 58-59 è possibile vedere le LUT per il calcolo di φ_d e φ_q. È stato fatto lo stesso procedimento di inserimento delle grandezze all'interno delle LUT per il calcolo di: $C_m, \frac{\partial \varphi_d}{\partial i_d}, \frac{\partial \varphi_d}{\partial i_q}, \frac{\partial \varphi_q}{\partial i_d}, \frac{\partial \varphi_q}{\partial i_q}$.

Sistema meccanico

Il sistema meccanico è uguale a quello implementato per i modelli lineari (vedi figura 34).

3.2.3 Risultati simulazione modello LUT

I risultati delle simulazioni che verranno mostrati, saranno uguali per i modelli di motore non lineare con le seguenti specifiche:

- Velocità = 2500 rpm
- Coppia resistente = 0 Nm (a vuoto)

Verranno raffigurate rispettivamente per ambe due i modelli:

- tensioni di fase (filtrate con filtro passa basso);
- correnti di fase in riferimenti d-q;
- velocità;
- transitori (passaggio da 1500 a 2500 rpm);

Per poter raggiungere velocità più elevate come 2500 rpm è stata alzata la tensione del bus DC portandola da 20 V a 50V.

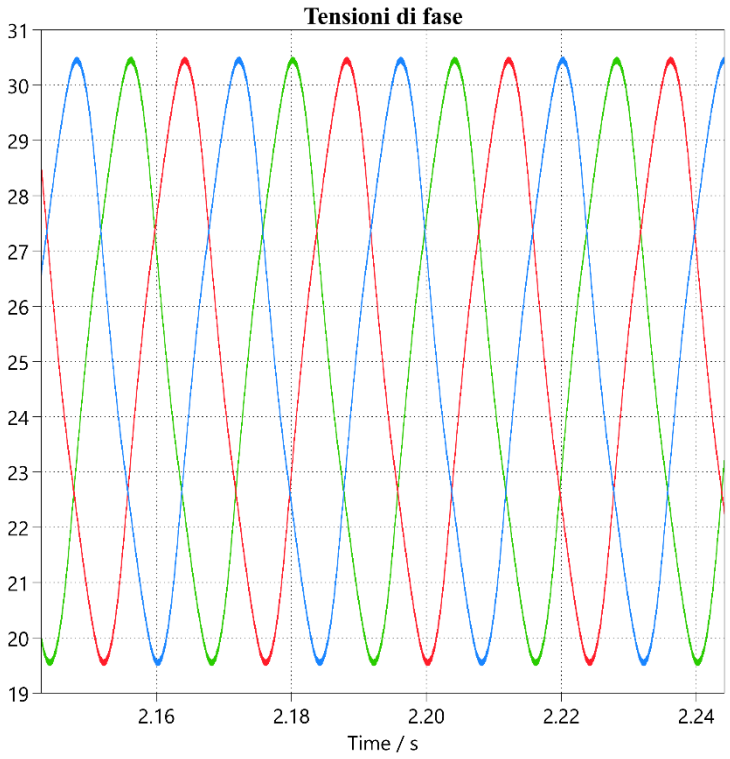


Figura 60: tensione di fase – modello LUT non lineare

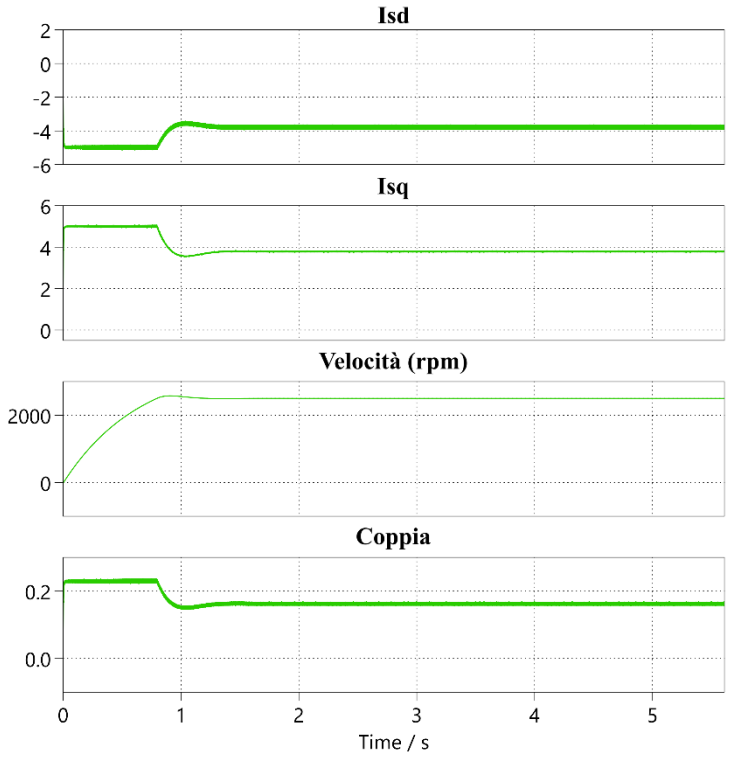


Figura 61: correnti di fase, velocità e coppia in simulazione PLECS – modello LUT non lineare

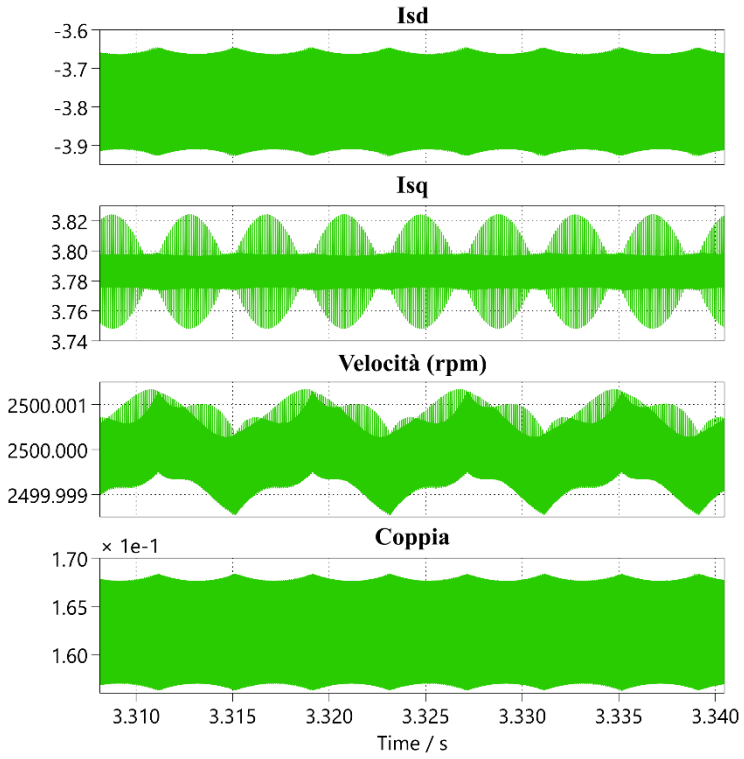


Figura 62: zoom figura

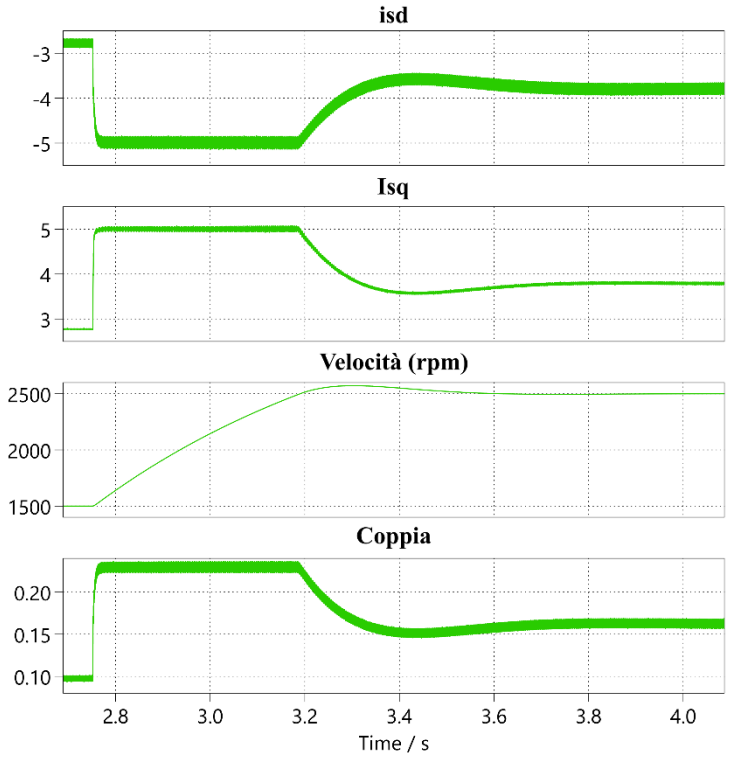


Figura 63: transitori 1500/2500 rpm – modello non lineare LUT

3.2.4 Modello flussi inversi

Come detto in precedenza, questo tipo di modello è stato realizzato come ulteriore verifica e confronto con il modello LUT. Una volta che i due modelli saranno validati e confrontati potranno essere portati su PHIL.

A livello di implementazione software per questo modello è stato modificato solamente il modello del motore lasciando interamente uguali tutte le altre parti di controllo e modellizzazione del sistema.

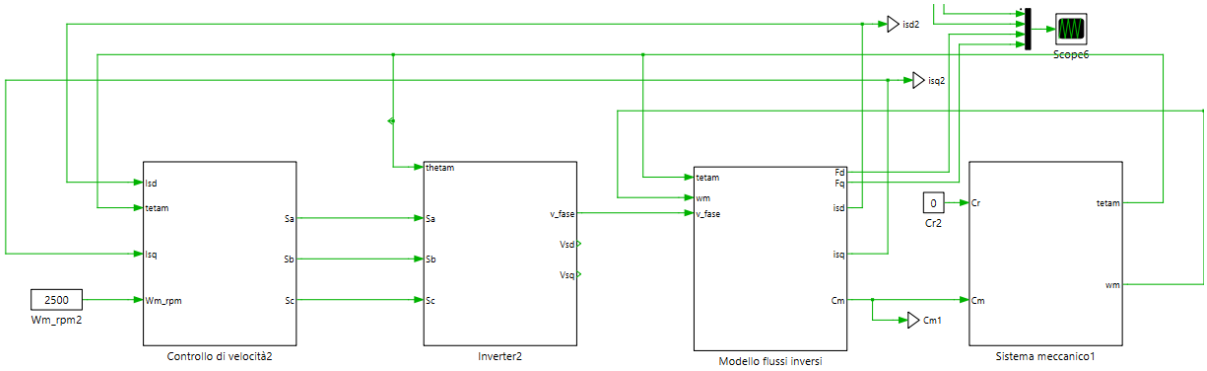


Figura 64: azionamento modello flussi inversi

Modello flussi inversi

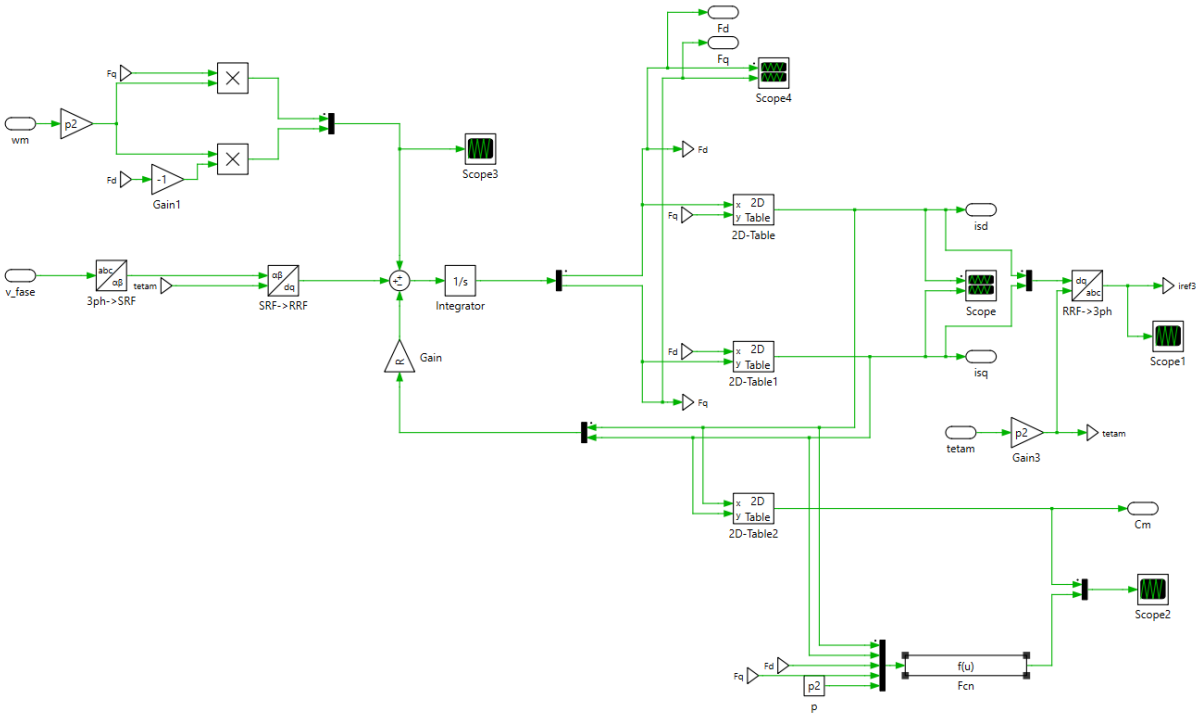


Figura 65: modello flussi inversi

Si nota che a differenza del modello LUT, il modello flussi inversi viene costruito in maniera opposta, ricevendo in ingresso i valori di flusso φ_d, φ_q e attraverso i blocchetti LUT restituisce i valori di corrente e coppia calcolati. Sono quindi stati ricalcolati e graficati gli andamenti dei flussi d e q inversi.

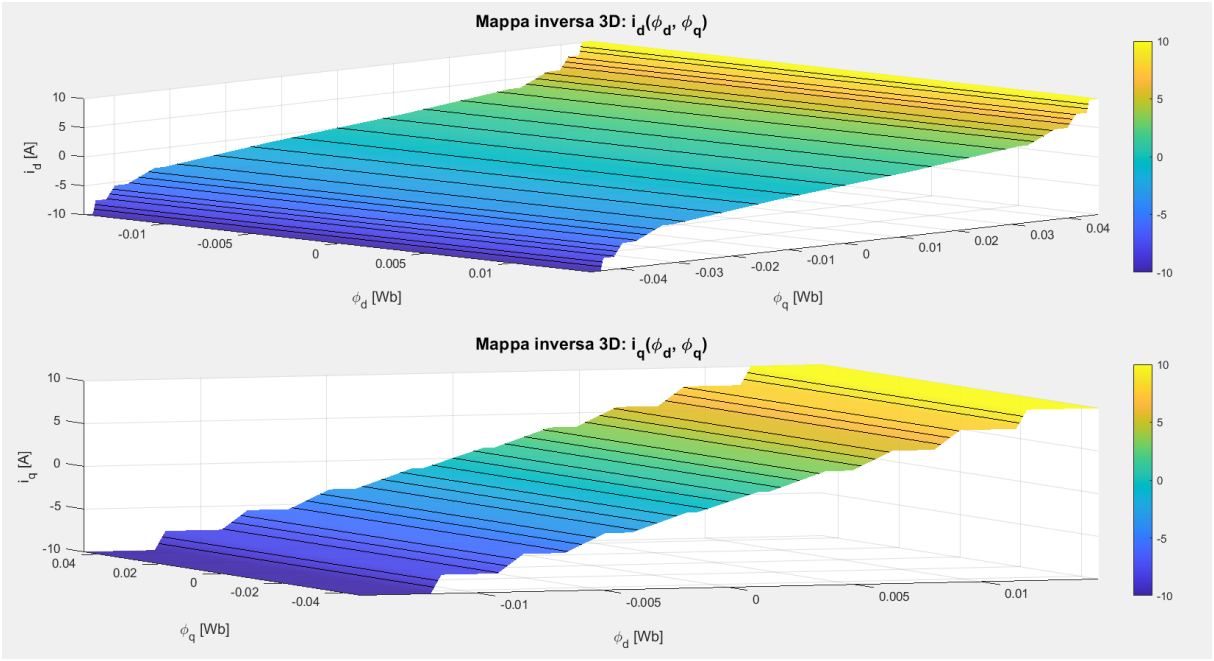


Figura 66: andamenti flussi inversi

Grazie a questo modello riusciremo quindi a verificare se il funzionamento delle modellizzazioni non lineari applicate, sia corretto o meno.

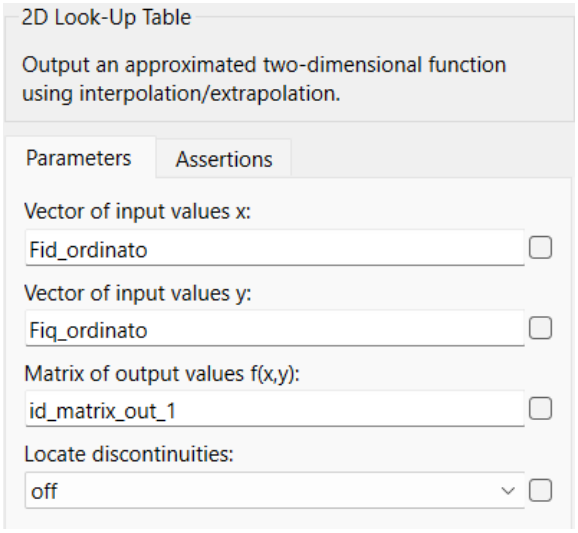


Figura 67: LUT i_{sd}

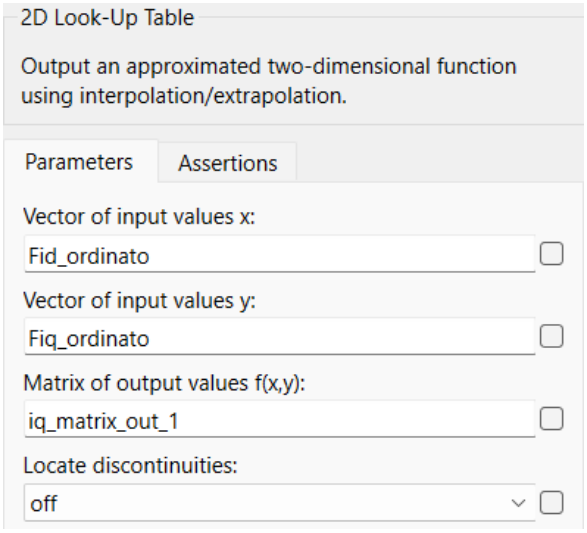


Figura 68: LUT i_{sq}

3.2.5 Risultati simulazione modello flussi inversi

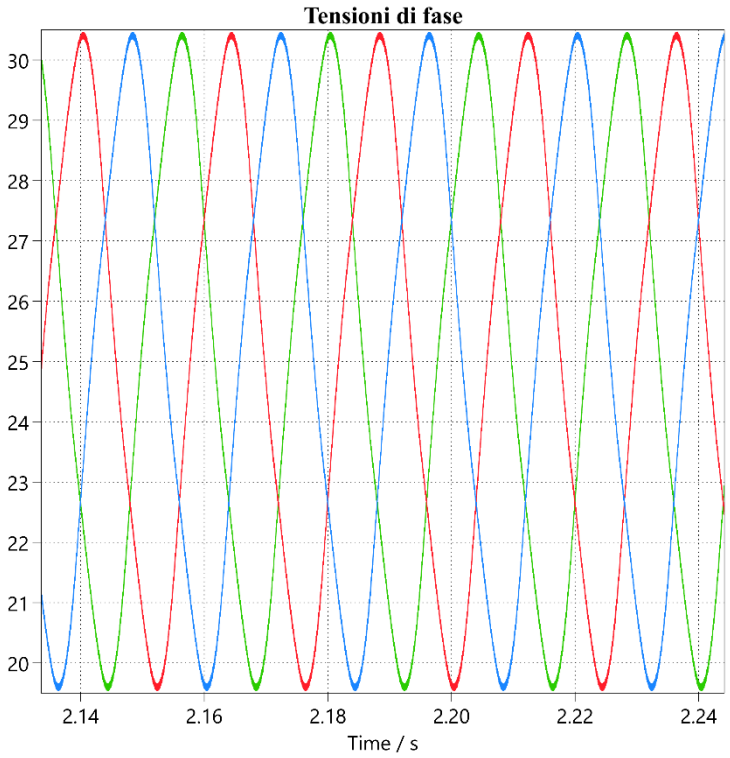


Figura 69: tensioni di fase – modello flussi inversi

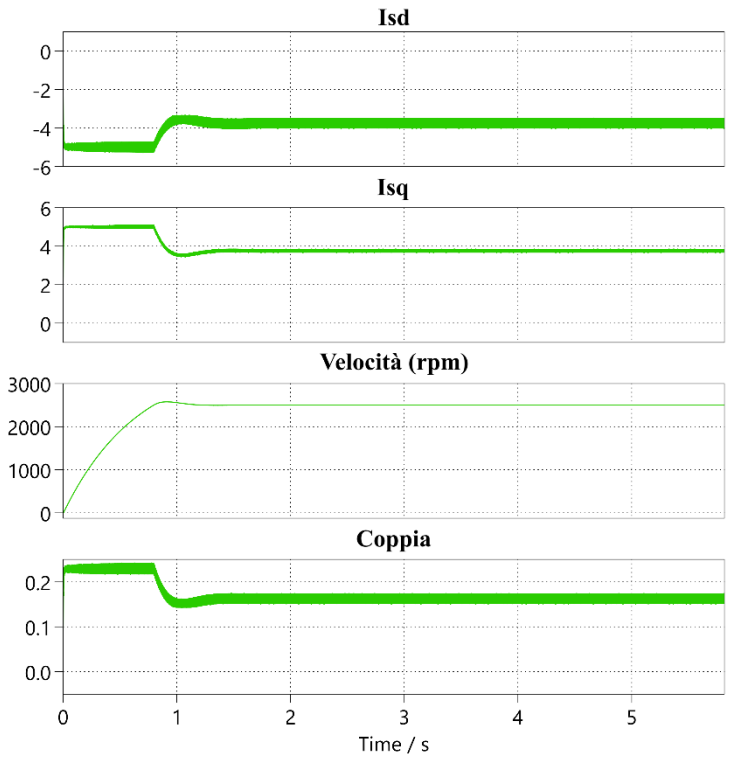


Figura 70: andamenti di corrente, coppia e velocità – modello flussi inversi

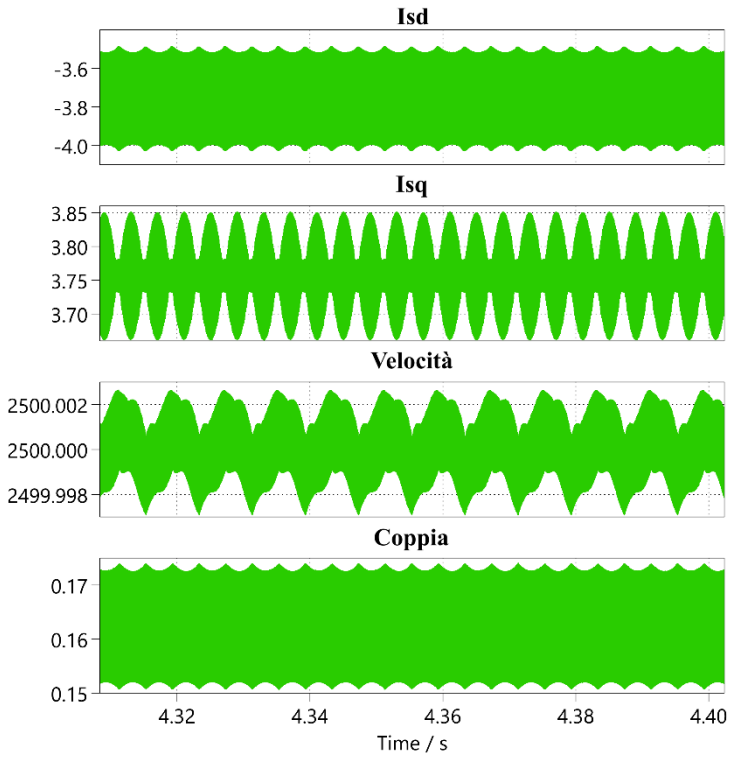


Figura 71: zoom figura 70

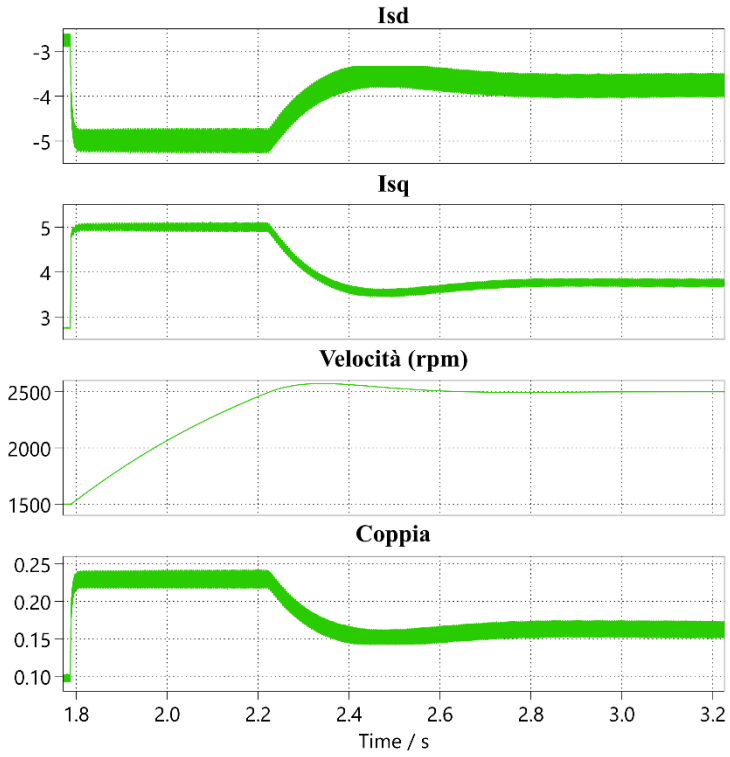


Figura 72: transitori 1500/2500 rpm – modello flussi inversi

Dalle figure mostrate si può vedere come i due modelli abbaino lo stesso comportamento nelle condizioni di funzionamento descritte in precedenza. Per avere una visione più chiara e certa della

veridicità delle simulazioni dei modelli non lineari, sono stati realizzati degli scope che mettono a confronto le varie grandezze di entrambi i modelli.

Confronto modelli non lineari

Sono state confrontate le stesse grandezze mostrate in precedenza così da avere un riscontro chiaro ed efficacie delle simulazioni

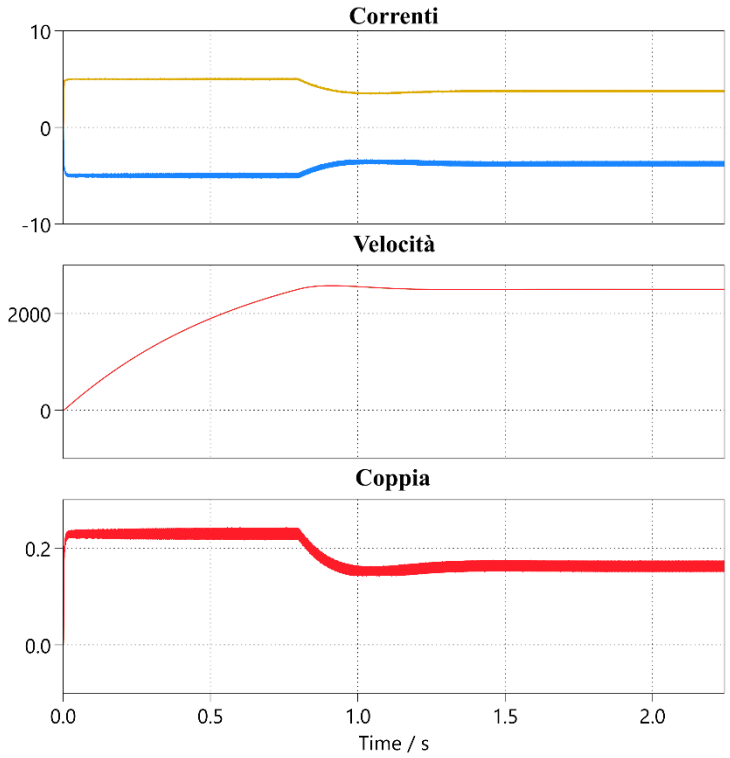


Figura 73: confronto correnti, velocità e coppia dei due modelli non lineari

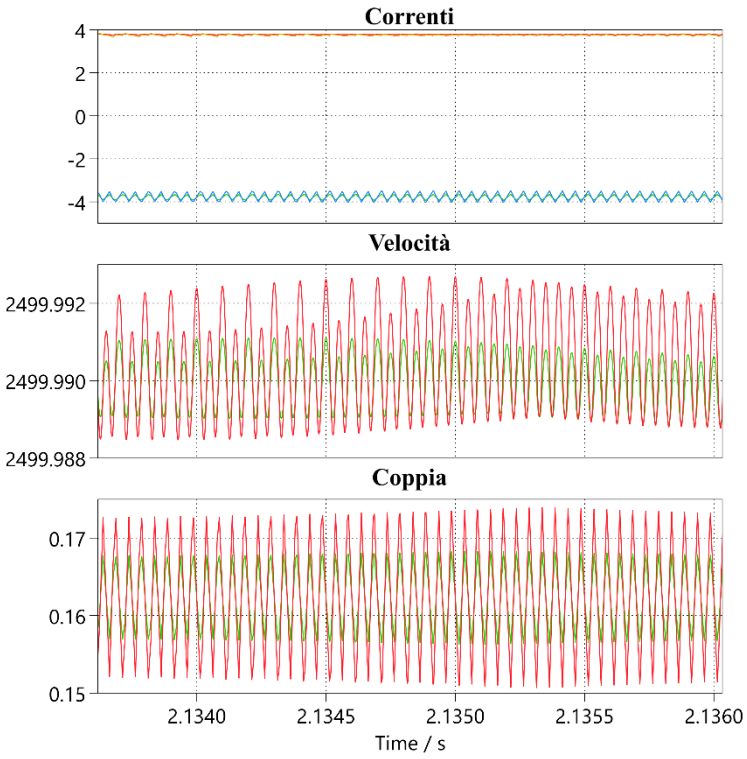


Figura 74: zoom figura 73

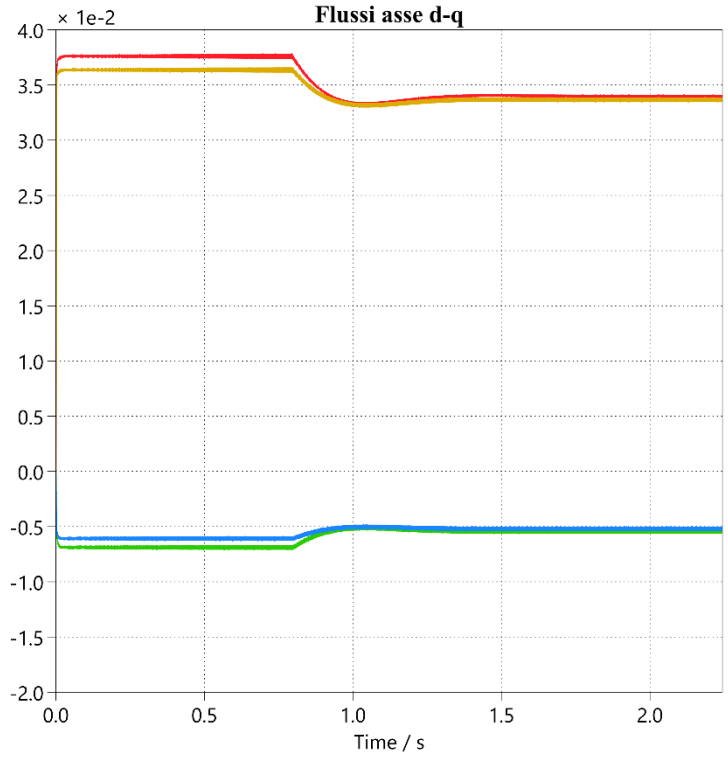


Figura 75: andamento flussi d-q dei modelli non lineari

I risultati delle simulazioni (sia lineari che non lineari) risultano quindi verificati e funzionanti; sarà quindi ora possibile implementare il tutto sull'architettura PHIL verificando di ottenere i risultati osservati fino ad ora.

Capitolo 4

Implementazione su Architettura Power Hardware in the Loop (PHIL)

In questo capitolo viene descritto l'apparato strumentale impiegato per l'implementazione in ambiente Power Hardware-in-the-Loop (PHIL) dei modelli lineari e non lineari sviluppati e analizzati nei capitoli precedenti. Dopo aver definito e verificato il comportamento dei modelli in simulazione pura tramite PLECS, è stato infatti possibile trasferire tali modelli in un contesto sperimentale ibrido, nel quale una parte del sistema è eseguita in tempo reale e interagisce con dispositivi fisici tramite un'interfaccia di potenza. L'obiettivo del PHIL è riprodurre fedelmente il comportamento dinamico del modello simulato quando questo è sottoposto a condizioni operative reali, permettendo così di validare algoritmi di controllo, strutture di conversione e fenomeni elettromeccanici in presenza di componenti fisici. Per ottenere tutto questo è necessario un insieme coordinato di strumenti hardware e software in grado di garantire:

- esecuzione real-time del modello elettrico tramite processore dedicato o real-time target;
- conversione analogico-digitale/ digitale-analogico dei segnali di misura e comando;
- interfaccia di potenza bidirezionale per applicare al dispositivo sotto test correnti o tensioni in tempo reale;

Nel seguito del capitolo verranno quindi presentati in modo dettagliato:

- l'architettura generale del banco PHIL;
- il processore o simulatore real-time utilizzato per l'esecuzione dei modelli (e la loro esportazione da PLECS/Simulink);
- i convertitori A/D e D/A e i moduli di I/O necessari allo scambio dati;
- l'interfaccia di potenza e i dispositivi fisici coinvolti nelle prove;
- la procedura di implementazione, parametrizzazione e verifica del modello in real-time.

Questa analisi consente di collegare la fase di modellazione e simulazione sviluppata nei capitoli precedenti con l'attività sperimentale vera e propria, fornendo il contesto tecnico nel quale i modelli lineari e non lineari sono stati validati attraverso prove PHIL.

4.1 Realizzazione dei software emulatore e DUT

Come detto nel capitolo 1, l'architettura Power Hardware in the Loop (PHIL) ha origine dal concetto di Hardware in the Loop, con il quale si indica lo scambio di segnali in loop fra un sistema fisico reale sotto test (detto *device under test,* DUT) e un modello simulato capace di emulare il comportamento di un sistema elettrico qualsiasi (Emulatore).

I due sistemi dovranno quindi essere realizzati in modo da poter comunicare tra loro, inseguendo i riferimenti e le informazioni che si scambiano. La realizzazione di essi è stata implementata su due software differenti:

- DUT – implementazione del controllo su Matlab Simulink;
- Emulatore – implementazione controllo e modelli motore su PLECS;

I due sistemi comunicheranno tra loro grazie ad una serie di dispositivi reali che si occuperanno di scambiare informazioni e potenza tra di loro.

4.1.1 Realizzazione e controllo Emulatore

Il controllo dell'emulatore, come detto in precedenza, è implementato sul software Plecs comunicante con il circuito reale tramite la RT BOX. L'interfaccia di controllo (DASHBOARD), nella figura 76 a seguire, presenta varie sezioni di controllo e monitoraggio del sistema.

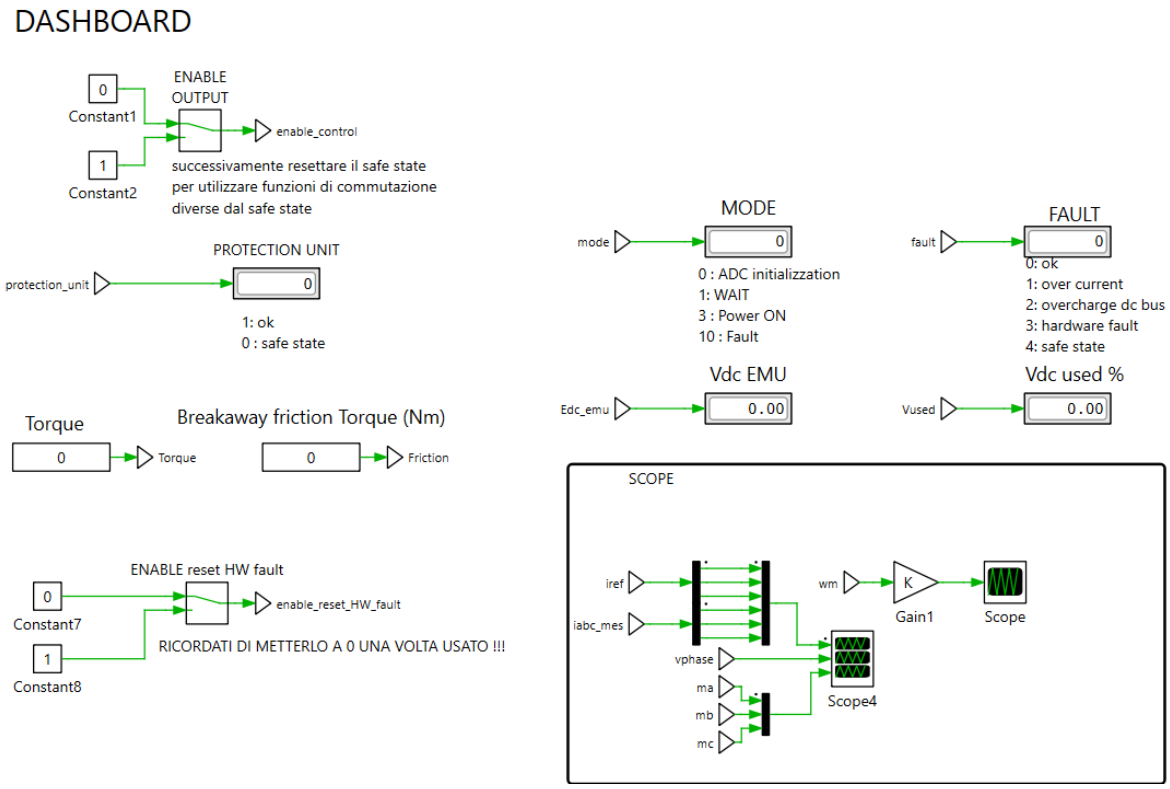


Figura 76: Dashboard di controllo e monitoraggio emulatore in Plecs

Descrizione della dashboard:

- ENABLE OUTPUT: è il comando di avvio del software che tramite uno switch manderà il segnale di enable control ai PI del controllo di corrente;
- PROTECTION UNIT: una volta inviato il segnale di enable il sistema entrerà nella condizione di "safe state", ovvero, uno stato di sicurezza in cui l'inverter emulatore non risponderà alle funzioni di commutazione che gli vengono fornite. Per uscire da questa condizione è stato inserito un semplice interruttore che all'invio dell'impulso permetterà di switchare lo stato di protection unit in 1 = OK.

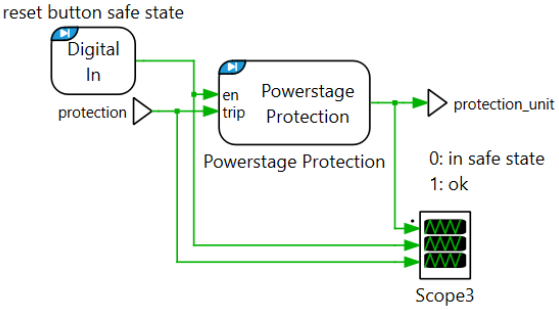


Figura 77:implementazione safe state

-Risulta opportuno precisare che i blocchi contraddistinti da questa icona  *identificano le interfacce hardware che consentono al modello di comunicare con la RT Box fisica, permettendo lo scambio di segnali attraverso i bus di ingresso e uscita-*

- SEGNALI DI FAULT: prima di procedere all'avvio del software è importante controllare i display relativi ai possibili fault; dalla fig. 76 si possono vedere cinque possibili stati:
  - 0 → *ok*: tutto funzionante;
  - 1 → *overcurrent*: il sistema vede una corrente maggiore della I_{MAX} (10 A);
  - 2 → *overcharge bus DC*: il sistema vede una tensione maggiore della V_{MAX} (40 V);
  - 3 → *hardware fault*: errori derivanti dalla scheda di segnale;
  - 4 → *safe state*: condizione in cui si è parlato precedentemente.

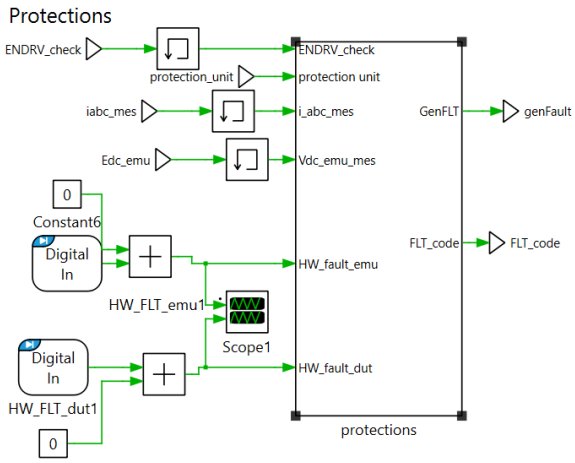


Figura 78: implementazione protezioni per segnali di fault

- DISPLAY DI MONITORAGGIO: monitoro coppia, coppia frenante e alimentazioni.
- SCOPE DI CONTROLLO DEL SISTEMA: controllo delle correnti di fase e modulanti.

Una volta che si avrà lo stato di OK (protection unit=1, mode=3, fault=0), il software sarà pronto per funzionare e inizierà a leggere le correnti misurate nel circuito reale per poi darle al modello della macchina, che permetterà di avviare il controllo di corrente dell'emulatore come si vede in fig. 79.

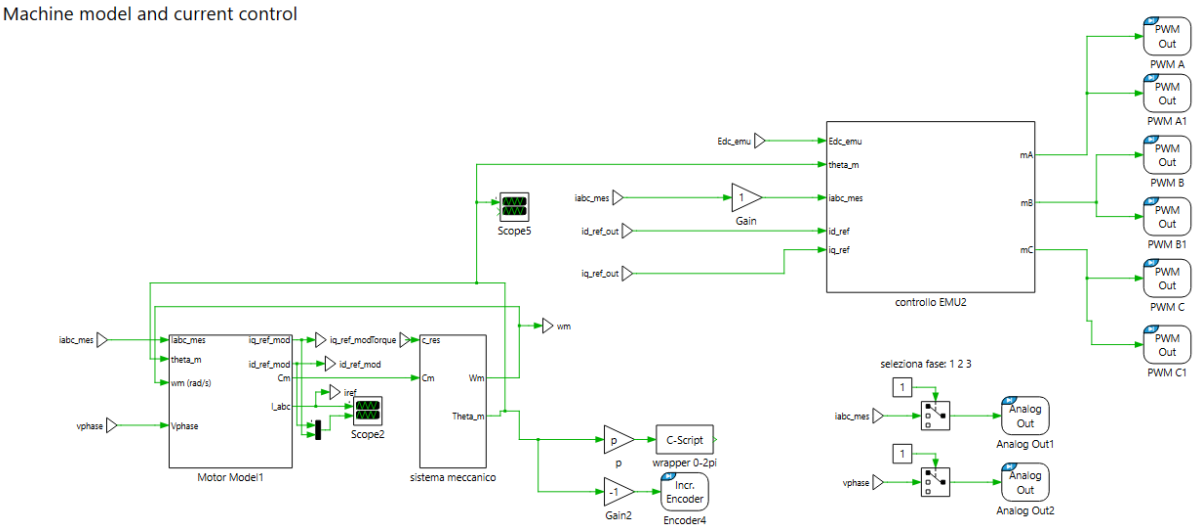


Figura 79: modello della macchina e controllo di corrente

Il controllo di corrente implementato si preoccuperà di inseguire le correnti misurate dalla scheda di segnale nel circuito reale (iabc_mes) e generare delle modulanti che permetteranno di pilotare l'inverter emulatore del modello macchina richiesto.

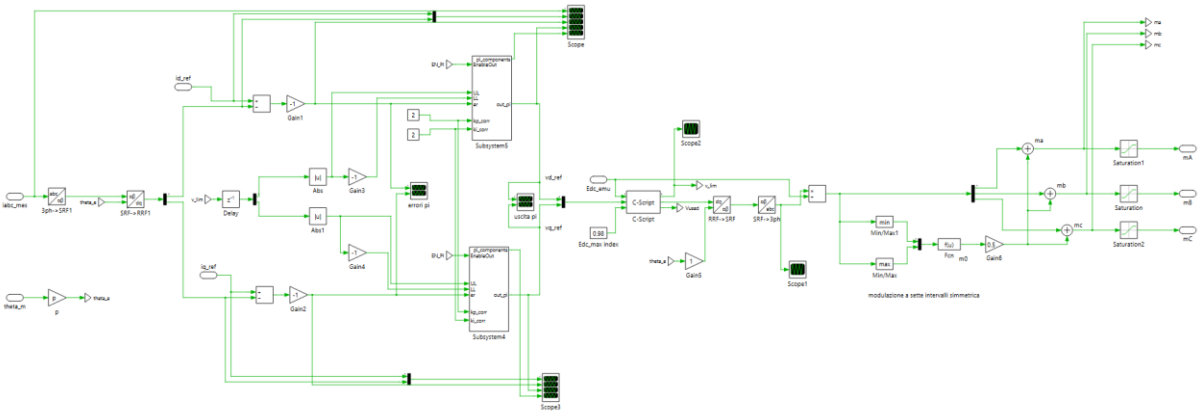


Figura 80: controllo di corrente dell'emulatore con modulazione a 7 intervalli

La regolazione dei PI discreti è stata fatta considerando:

- $K_i = 19.25$
- $K_p = \left(\frac{\mathrm{L_{emu}}}{\mathrm{R_{emu}}}\right)\mathrm{K_i}$ *dove* $L_{emu} = 1800^{-6}\ H$ *e* $R_{emu} = 41^{-3}\Omega$
- $f_s = \frac{1}{T_{samp}}$ *dove* $T_{samp} = 20^{-6}$

È stato implementato il controllo dell'inverter sempre con la modulazione 7 intervalli spiegata in precedenza, con la conseguente creazione delle modulanti m_a, m_b, m_c.

Misurazioni segnali di corrente reali

I valori di corrente e tensione richiesti dal controllo di velocità del DUT vengono letti e digitalizzati dalla parte di misura implementata sul software emulatore. Tramite i blocchi "Analog in" vengono raccolti i segnali di corrente e tensione e vengono mandati all'interno del sottosistema "measurment", dove i vari segnali verranno convertiti per essere utilizzati nel controllo di corrente poc'anzi mostrato.

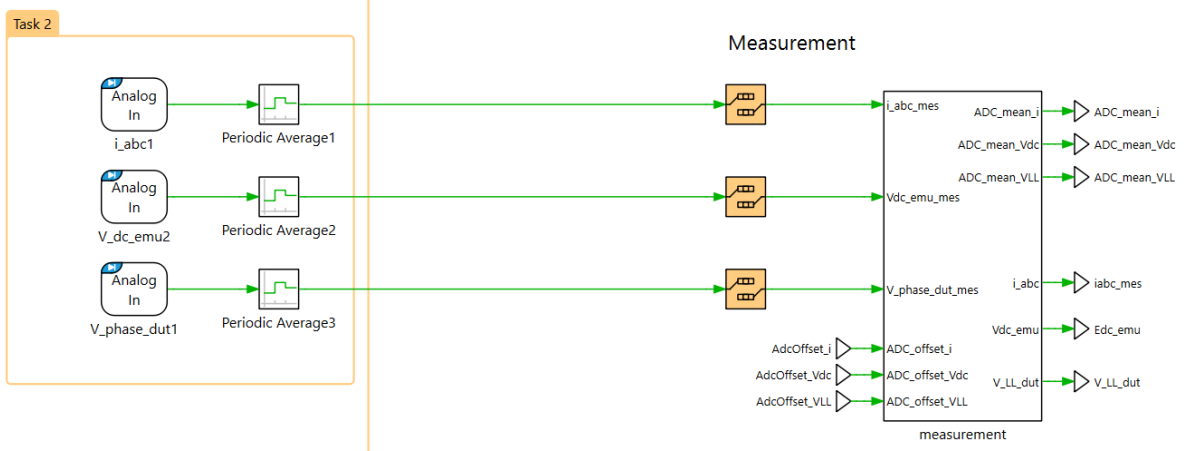


Figura 81: misurazione segnali reali

Macchina a stati

Il coordinamento tra i vari stati del software durante il funzionamento è regolato dalla "Macchina a stati o State machine". Una macchina a stati è un modello concettuale utilizzato per progettare sistemi di controllo che evolvono attraverso una serie di stati in risposta a degli input esterni. Questo approccio consente di descrivere il comportamento di un sistema in termini di transizioni tra stati specifici, ognuno dei quali rappresenta una particolare configurazione del sistema. In altre parole, è un metodo strutturato per gestire la logica complessa e le decisioni basate sugli eventi.

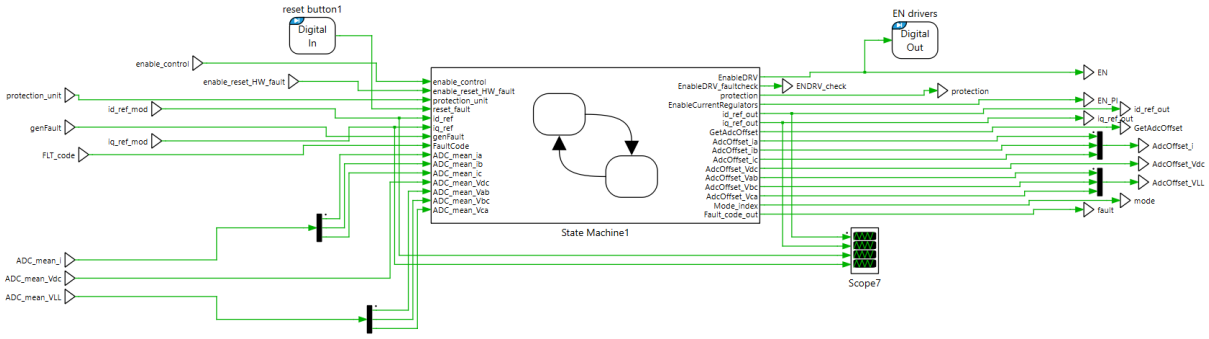


Figura 82: state machine controllo emulatore

Il flow chart implementato è mostrato in figura 83. Il punto di ingresso è il pallino nero, nel primo stato si rileva l'offset presente sui canali e si passa al secondo stato quando è passato un secondo dall'accensione. Questo è lo stato di attesa, tutti i riferimenti sono posti a zero, i controlli disattivati e gli offset imposti sulle uscite. Quando sulla dashboard si preme il bottone start, il controllo si mette in uno stato in cui è ancora tutto disabilitato ma può partire in qualsiasi momento. Se dalla dashboard si

attiva il controllo, i regolatori vengono attivati e l'azionamento è messo in funzione. Se, in un qualsiasi momento, viene rilevato un fault le correnti sono poste a zero e il driver esterno spento. Il sistema, se non ci sono più le condizioni di fault, esce dalla modalità di protezione quando viene premuto il tasto "RESET FAULT" sulla dashboard.

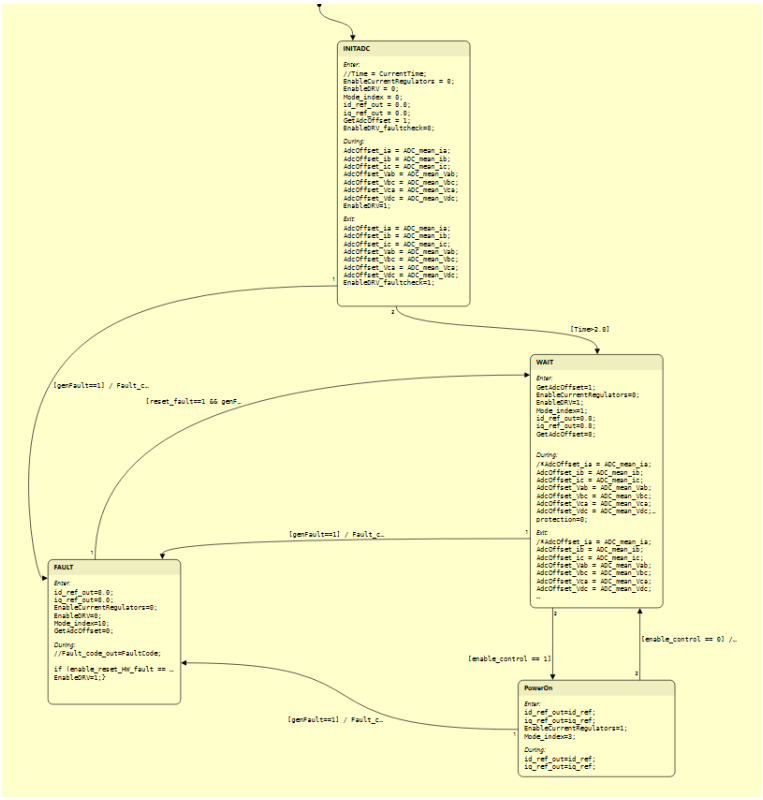


Figura 83: flow chart della macchina a stati

4.2 Setup sperimentale

Dovendo emulare le condizioni operative di un motore sincrono trifase si deve analizzare il modello circuitale, in cui ogni fase è descritta dalla serie di una resistenza, un'induttanza e un generatore di tensione che rappresenta le forze elettromotrici (fem). Il modello può essere replicato nel sistema PHIL in quanto l'accoppiamento rappresenterà la serie di resistenze e induttanze, mentre l'inverter emulatore la fem del motore. In un inverter non è possibile formare fisicamente un collegamento delle fasi a stella, ma tramite il controllo si può ottenere un cortocircuito virtuale imponendo il vincolo di somma nulla delle correnti.

Il controllo è operato da un processore montato su una scheda di sviluppo della Texas Instruments, la Delfino F28379D (vedi Fig. 85) mentre gli inverter sono moduli di espansione della scheda di sviluppo, denominati BOOSTXL-DRV8305EVM, e sono uguali (vedi Fig. 84).

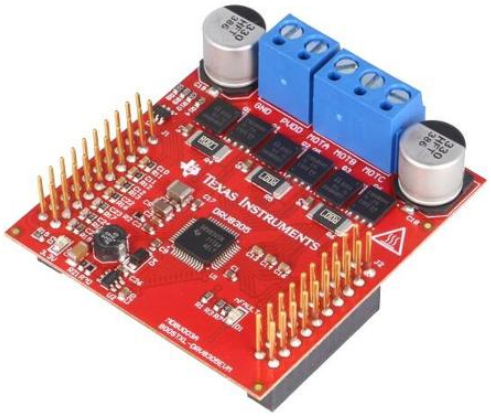


Figura 84: Modulo di espansione BOOSTXL-DRV8305EVM

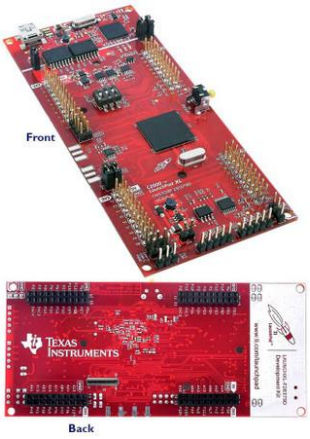


Figura 85: Scheda di sviluppo Delfino F28379D

I due convertitori devono essere totalmente isolati l'uno dall'altro per evitare il ricircolo di correnti dovute ad una differenza di tensione che si vedrebbe fra i due poli negativi dei bus DC in ingresso ad essi. Studiando la loro configurazione interna, si nota che i due moduli non sono full isolated, ovvero non vi è un completo isolamento tra i segnali a bassa tensione provenienti dal controllo e quelli di potenza. Il ground del livello segnale è direttamente connesso al polo negativo dell'alimentazione.

Infine, risulta importante precisare come DUT ed emulatore vengano controllati:

- Il DUT viene comandato dal controllo di velocità implementato sul software Matlab Simulink in cui sarà possibile scegliere il riferimento da inseguire. In questo caso la scheda di sviluppo Delfino comunicherà con il software per poi comandare il modulo di espansione (vedi Fig. 86).

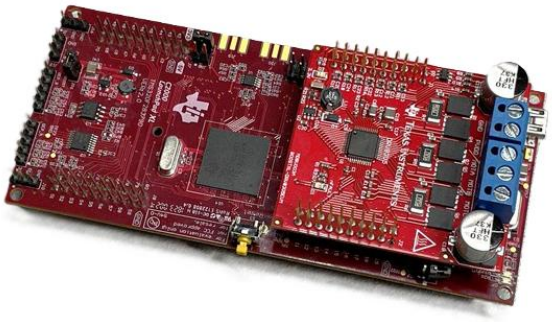


Figura 86: Scheda di sviluppo delfino con modulo di espansione

- L'emulatore invece è il sistema composto dallo stesso convertitore rappresentato in fig. 84 ma installato su una scheda di segnale che ha il compito di misurare tensioni e correnti (vedi figura 88). Il suo controllo è invece implementato sul software Plecs, comunicante con un simulatore real time chiamato RT-BOX 3 (vedi Fig. 87).

Figura 87: Simulatore real time RT-BOX 3

Scheda di condizionamento del segnale

Allo scopo di evitare la circolazione di correnti indesiderate tra i poli negativi dei convertitori, è stato implementato un sistema di isolamento dei segnali, mantenendo il DUT sulla scheda di sviluppo e isolando le comunicazioni con l'EMU. Il circuito prevede sensori ad effetto Hall per la misura di corrente, sensori di tensione e uno stadio di alimentazione che, tramite un Buck da 12 V, genera le tensioni di riferimento a 5 V e 3,3 V per i componenti integrati. L'isolamento dei segnali PWM e di fault è ottenuto con isolatori unidirezionali alimentati da entrambe le sezioni del sistema. La misura delle correnti avviene su ogni fase mediante sensori ACS722, con la possibilità di monitorare le grandezze principali tramite connettori dedicati e bufferizzazione con amplificatori operazionali.

Figura 88: scheda di condizionamento dei segnali

Per avere una visione più chiara di tutto quello che è l'architettura spiegata fino ad ora viene in aiuto la figura seguente in cui è possibile vedere la postazione completa dove sono state effettuate tutte le prove.

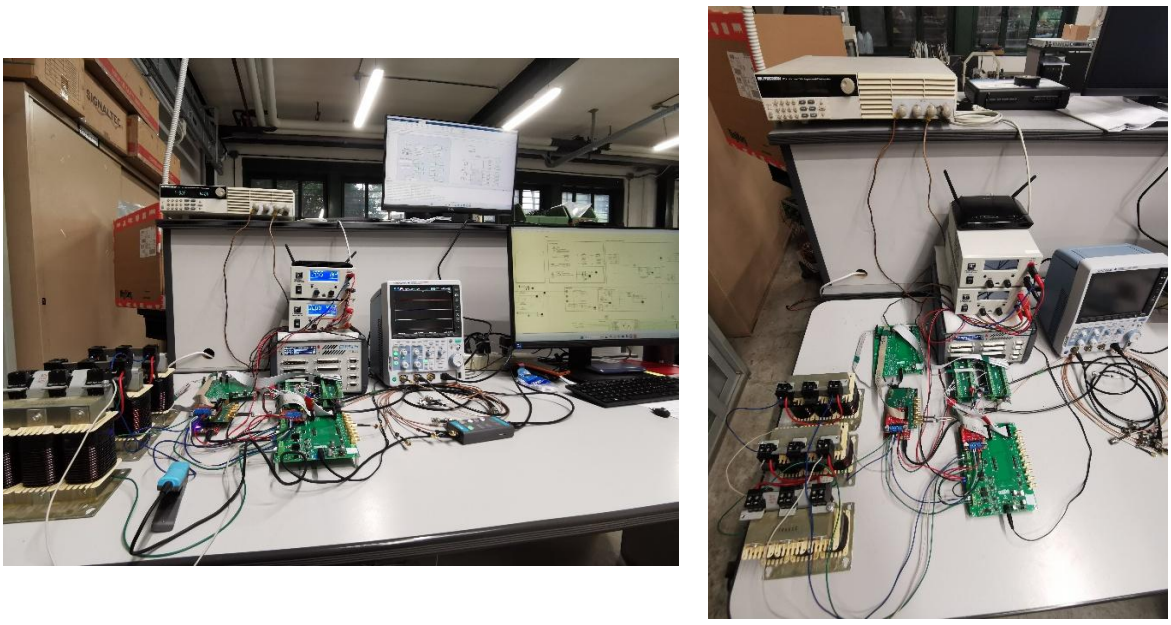


Figura 89: Architettura PHIL implementata in laboratorio

Nella figura di sinistra si possono osservare i due desktop utili per controllare e monitorare DUT ed emulatore rispettivamente con Matlab Simulink e Plecs. In quella di destra invece si notano meglio quelli che sono i componenti circuitali ed i collegamenti; si noti che, come resistori per la parte di potenza, è stato utilizzato un banco di resistori settato a circa 10 Ω in serie ai tre induttori

4.2.1 Realizzazione e controllo DUT

Il controllo del DUT, come detto in precedenza, è implementato su Matlab Simulink e segue in tutto e per tutto la costruzione del modello realizzata su PLECS per l'emulatore. Infatti, è presente un'interfaccia di monitoraggio e controllo proprio come l'emulatore (vedi Fig. 90).

Di seguito verranno mostrate quelle che sono le differenze rispetto all'emulatore:

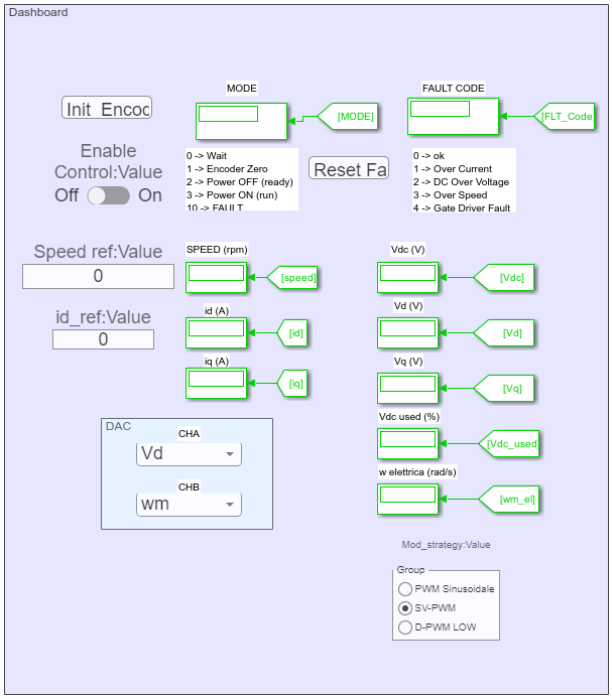


Figura 90:Interfaccia Simulink per controllo DUT

È stata realizzata una dashboard di controllo e monitoraggio del DUT in cui è possibile utilizzare diversi comandi e monitorare in tempo reale le varie grandezze di interesse.

- Inizializzare l'encoder su Simulink tramite l'apposito comando: "init_encoder" è necessario per attivare la procedura di inizializzazione dell'encoder, cioè, trovare lo zero del sistema di riferimento. Consiste nell'imporre una corrente costante lungo un'asse d o q per potere fare ruotare il rotore portandosi nella condizione di minore riluttanza e l'encoder ci darà un valore di posizione che diventerà il nostro 0. In questo modo conosceremo la posizione esatta del nostro rotore.
- Il comando di "enable" abilita i blocchi del controllo.
- Sui due display in alto sono visualizzati la modalità in cui sta funzionando il dispositivo e il tipo di errore, se presente.
- Il pulsante di reset permette di uscire dalla condizione di fault e, nel caso non siano più presenti errori, ritornare al funzionamento normale.
- Nella casella "SPEED REF" viene impostata la velocità di riferimento, che può essere cambiata anche mentre il dispositivo è in funzione.
- Vi sono poi una serie di display che mostrano le principali variabile del DUT, ma siccome sarebbe troppo complesso per il sistema aggiornare quei valori alla frequenza degli inverter,

hanno un tempo di campionamento di 0.8 s. Nei menù a tendina è possibile settare quali parametri visualizzare sui canali del DAC.

Il seguente sistema triggerato è stato realizzato per implementare tutta la parte di controllo e conversione utile all'interfacciamento con il sistema reale.

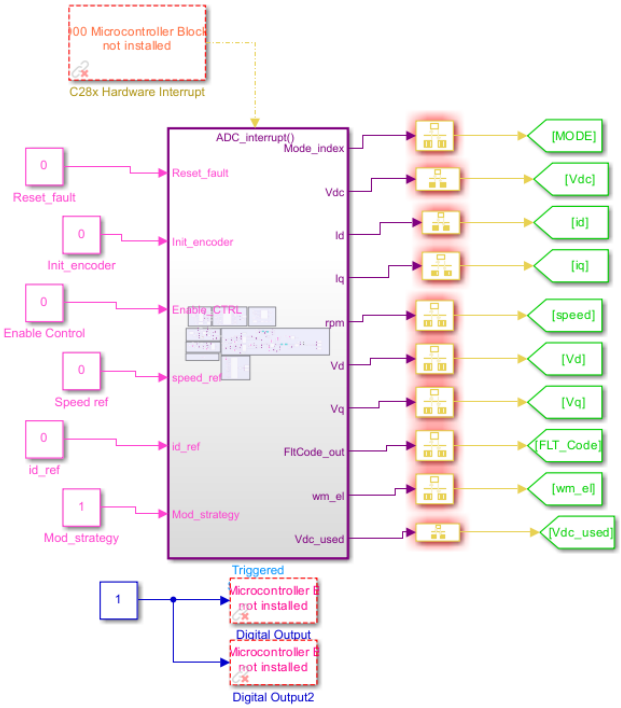


Figura 91: subsystem triggered controllo DUT

Conversione ADC

All'interno del blocco triggerato dell'emulatore è presente la conversione analogico digitale come per l'emulatore su PLECS. Disponendo di quattro ADC e tredici canali per ognuno (più due canali comuni), sono state campionate le misure delle stellate, delle correnti e delle tensioni sui due bus DC. A seguito del campionamento vi è la fase di quantizzazione in cui l'ADC assegna a ogni campione un valore discreto selezionato da un insieme finito di possibili valori. Questo processo introduce un errore di quantizzazione, poiché il valore assegnato può non corrispondere esattamente al valore reale del campione. La risoluzione dell'ADC, espressa in bit, determina il numero di livelli discreti possibili e, di conseguenza, l'accuratezza della conversione. Quelli presenti sono a 12 bit, ovvero 4096 livelli che garantiscono una sensibilità di $\frac{3.3V}{4096} = 0.806\ mV$..

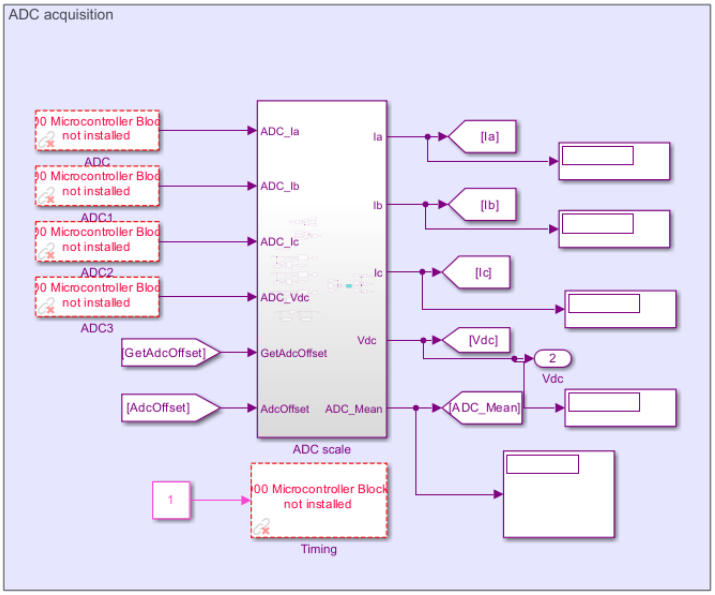


Figura 92: acquisizione misure

Controllo di velocità

Di seguito, in Fig. 93, viene illustrato il controllo di velocità implementato:

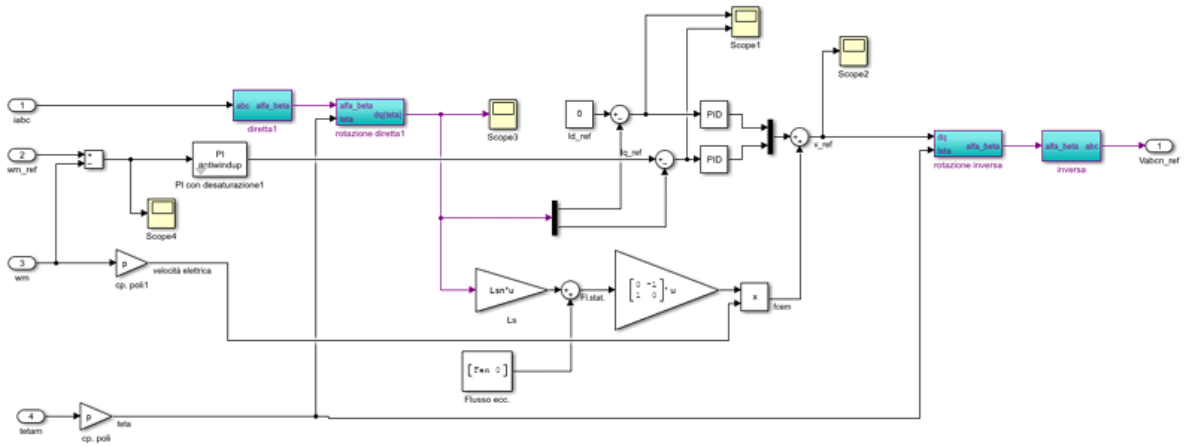


Figura 93: Controllo di velocità DUT

Per far sì che il controllo di velocità avvenga in modo corretto, è importante che il riferimento di tensione in uscita dal controllo venga inseguito; per farlo è stato implementato un controllo PWM sull'inverter DUT con tecnica di modulazione a sette intervalli simmetrica (spiegata in precedenza).

Si ricorda che a seconda del tipo di macchina utilizzato il controllo di velocità cambierà.

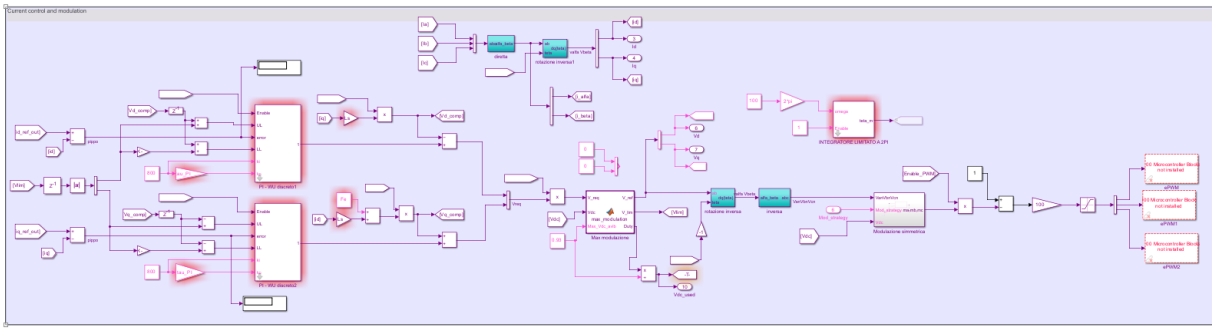


Figura 94: controllo velocità DUT per macchina brushless lineare

Macchina a stati

Come per il software implementato per l'emulatore anche in questo caso il tutto è coordinato da una macchina a stati:

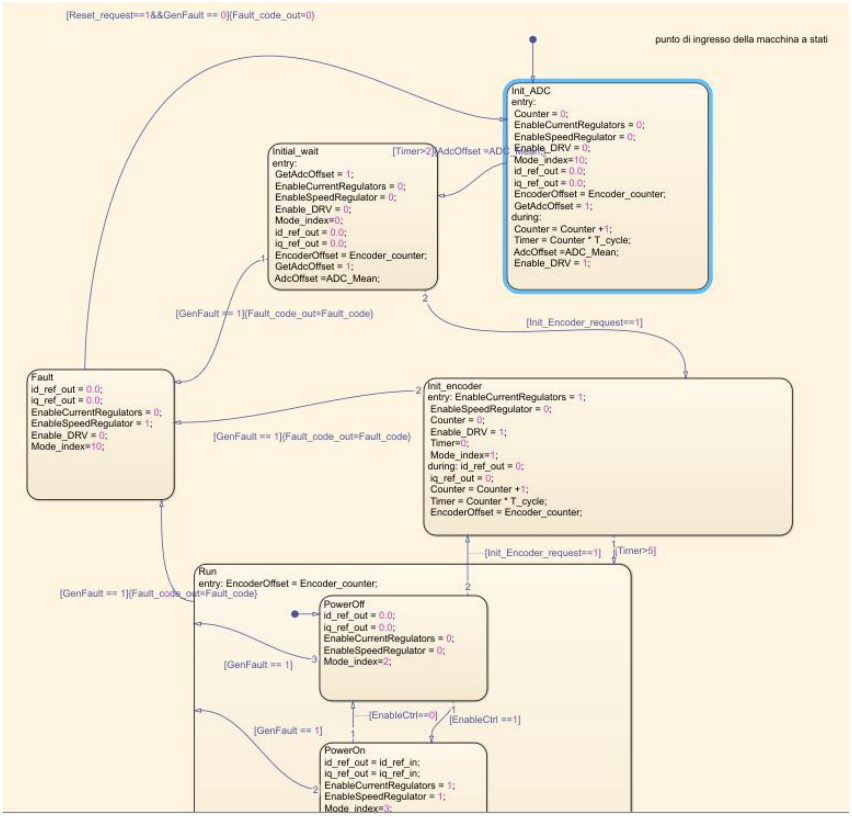


Figura 95: macchina a stati controllo DUT

4.3 Risultati sperimentali

Gli obiettivi di questa prova sperimentale sono:

- Verificare la corretta comunicazione tra DUT ed emulatore.
- Validare la stabilità e l'accuratezza dell'interfaccia di potenza.
- Dimostrare che il controllo implementato riuscisse a far seguire al DUT i riferimenti desiderati (velocità, correnti, tensioni).
- Confrontare il comportamento dei modelli di macchina emulati con quelli delle simulazioni fatte in precedenza e mostrate nel capitolo 3.

4.3.1 Procedura avviamento PHIL

Per verificare la validazione dei primi tre punti si è dovuto stilare quello che è il procedimento di avviamento dell'intero sistema per evitare di rimanere bloccati nei vari loop che possono venire a crearsi all'interno delle righe di codice delle state machine dei software:

1. Lanciare la simulazione sul software Matlab Simulink;
2. Collegare il programma Plecs alla RT-Box per poter comunicare con l'esterno;

A questo punto il sistema è interamente alimentato ed è pronto per essere avviato:

3. Resettare qualsiasi tipo di fault se sono presenti per entrambi i software tramite i rispettivi comandi:
    - su Simulink fare reset dal comando "Reset Fault" (vedi fig. 90);
    - su Plecs è stato inserito un piccolo pulsante sulla scheda esterna di comunicazione che invia il segnale logico;
4. Mandare comando di enable su Plecs, visibile sulla dashboard;
5. Inizializzare l'encoder su Simulink tramite l'apposito comando: "init_encoder" è necessario per attivare la procedura di inizializzazione dell'encoder, cioè, trovare lo zero del sistema di riferimento.

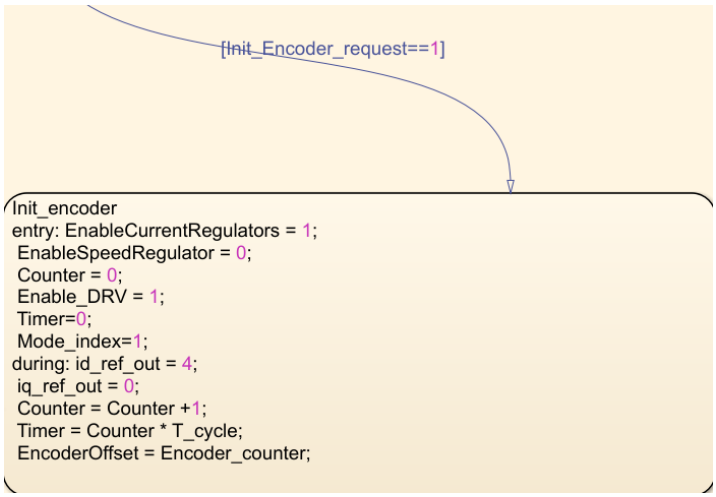


Figura 96:Inizializzazione encoder

6. Dare comando di enable su Simulink e impostare la velocità desiderata;

Rispettando tutte le procedure appena descritte secondo il seguente ordine capiremo se i due sistemi comunicheranno correttamente tra loro.

4.3.2 Implementazione dei modelli e risultati

Si può ora effettuare il confronto tra i due modelli per capire quali sono le differenze; si precisa che le simulazioni sono state lanciate mantenendo i parametri mostrati in precedenza per le simulazioni effettuate su PLECS:

Modelli lineari

- Velocità = 750 rpm
- Coppia = 0.2 Nm

Modelli non lineari

- Velocità = 2500 rpm
- Coppia = 0 Nm

Verranno raffigurate rispettivamente per i modelli:

- tensioni e correnti di fase;
- riferimenti di corrente;
- modulanti di controllo inverter;
- velocità;
- transitori;
- oneri computazionali;

Il tutto sarà confrontati con le simulazioni effettuate su PLECS nel capitolo 3 per verificarne la veridicità.

Per poter implementare i modelli all'interno del sistema PHIL basterà andare a riutilizzare quelli studiati nelle simulazioni precedenti e inserirli all'interno del software di controllo dell'emulatore (sempre su PLECS) che poi generare le modulanti di controllo per l'inverter EMU.

Blocco motore PMSM PLECS

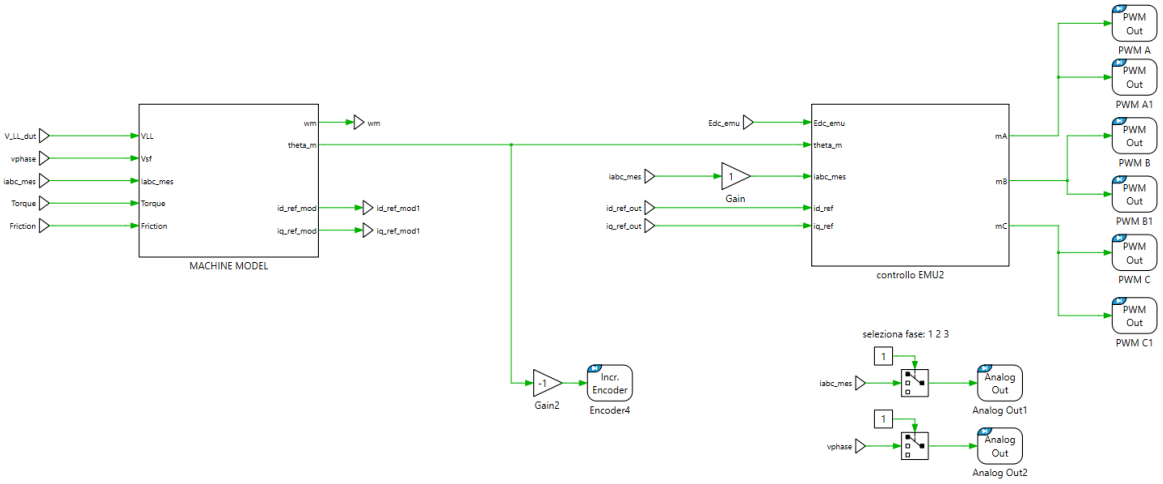


Figura 97: modello macchina PMSM implementato in PHIL

PHIL

SIMULAZIONE

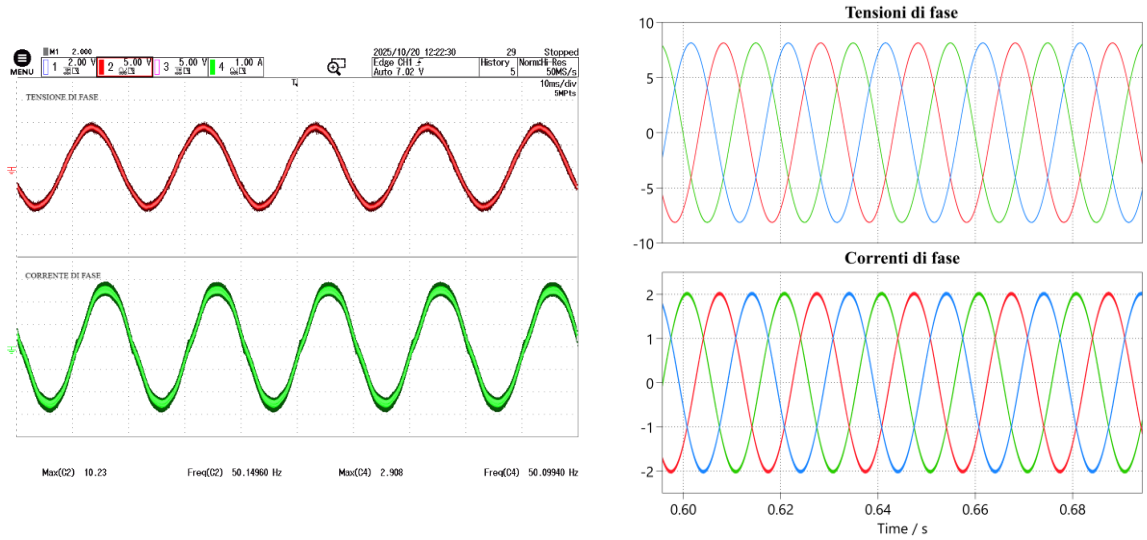


Figura 98:confronto tensioni e correnti di fase a regime, blocco motore Plecs – 750 rpm, 0.2 Nm

PHIL

SIMULAZIONI

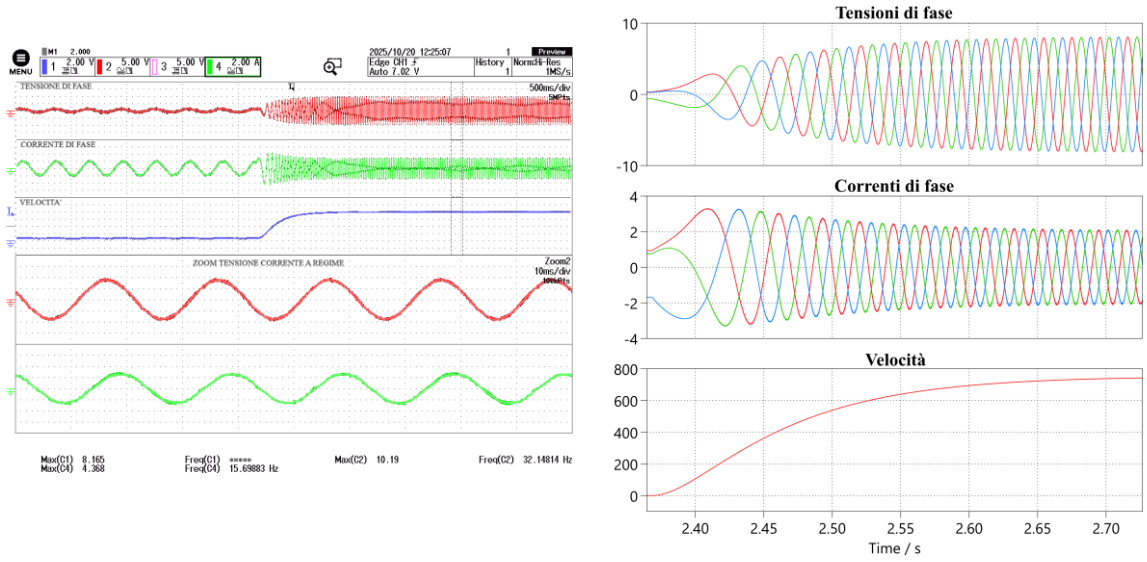


Figura 99: confronto transitori di corrente, tensione e velocità (10mV/rpm), blocco motore Plecs – 0-750 rpm

È possibile osservare come le forme d'onda di tensione e corrente misurate risultino perfettamente coincidenti con quanto visto con le simulazioni; anche i transitori risultano avere lo stesso identico comportamento.

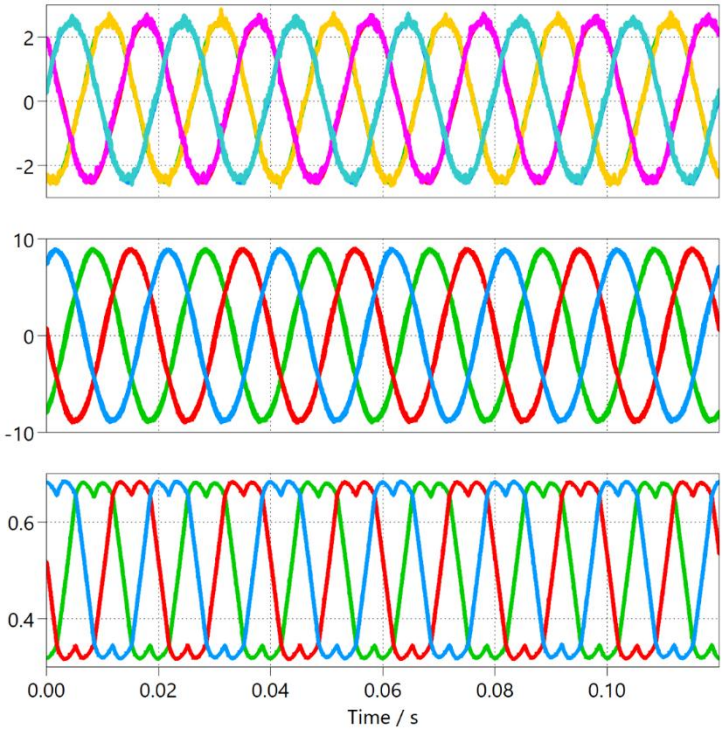


Figura 100: scope Plecs con correnti di fase e riferimenti, tensioni di fase e modulanti comando inverter – blocco motore Plecs

Modello matematico PMSM

PHIL SIMULAZIONI

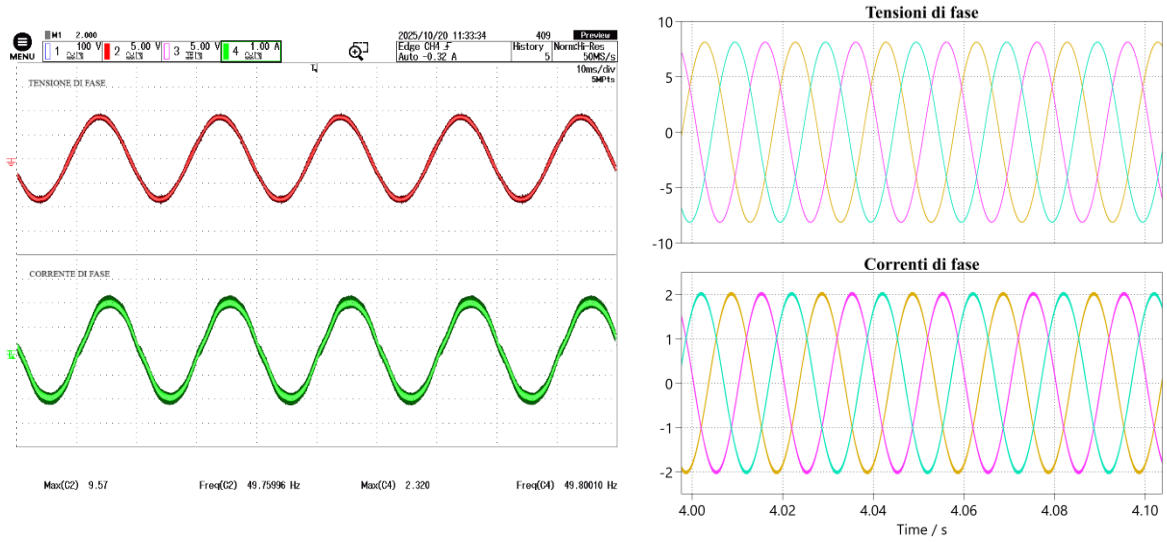


Figura 101: confronto tensione e corrente di fase a regime, modello matematico – 750 rpm, 0.2 Nm

PHIL SIMULAZIONI

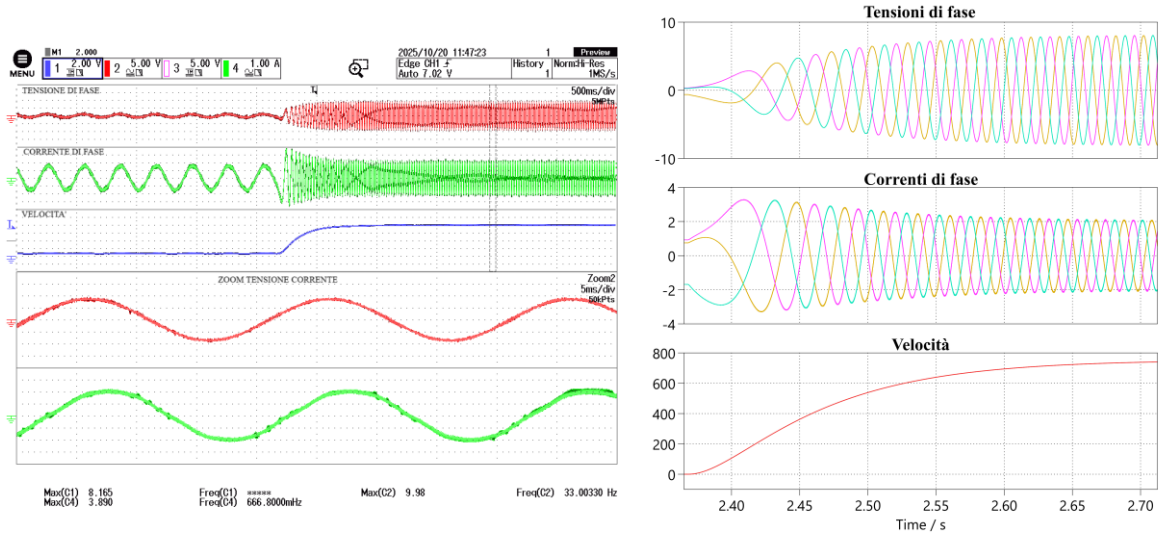


Figura 102 :confronto transitori di corrente, tensione e velocità (10mV/rpm), modello matematico – 0-750 rpm

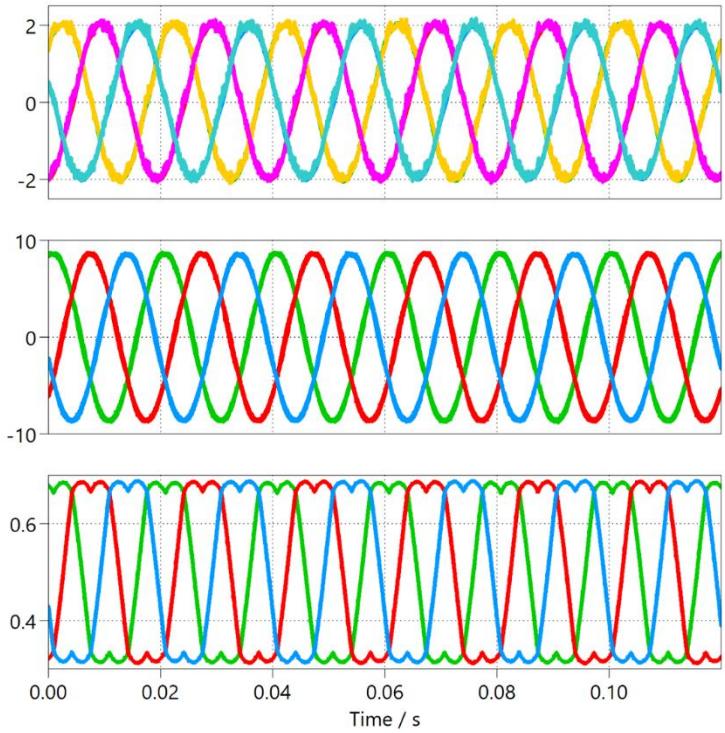


Figura 103: scope Plecs con correnti di fase e riferimenti, tensioni di fase e modulanti comando inverter – modello matematico

Dall'analisi dei grafici relativi alle differenti prove di simulazione si osserva una notevole sovrapposizione tra le grandezze elettriche e meccaniche simulate. Le forme d'onda di tensione e corrente di fase, risultano pressoché identiche nei due casi. Ciò conferma la coerenza del modello matematico sviluppato rispetto al modello di riferimento implementato nel blocco Plecs.

Durante la fase di avviamento, entrambi i modelli mostrano un comportamento dinamico analogo: l'aumento graduale della velocità, il picco iniziale della corrente dovuto alla richiesta di coppia e la successiva stabilizzazione in regime.

In conclusione, il confronto evidenzia una perfetta corrispondenza tra la simulazione offline in PLECS e l'architettura PHIL, validando il modello teorico proposto e l'efficacia della strategia di controllo adottata. Le simulazioni confermano che entrambi i sistemi forniscono risultati coerenti, stabili e affidabili, rendendo il modello matematico pienamente utilizzabile per ulteriori analisi e ottimizzazioni del sistema di azionamento. Un ultimo ulteriore confronto è stato effettuato su quello che è l'onere computazionale delle due simulazioni in PHIL.

Analisi onere computazionale

Per completare il confronto tra i due modelli, è stata condotta anche un'analisi dell'onere computazionale associato alla loro esecuzione in tempo reale.

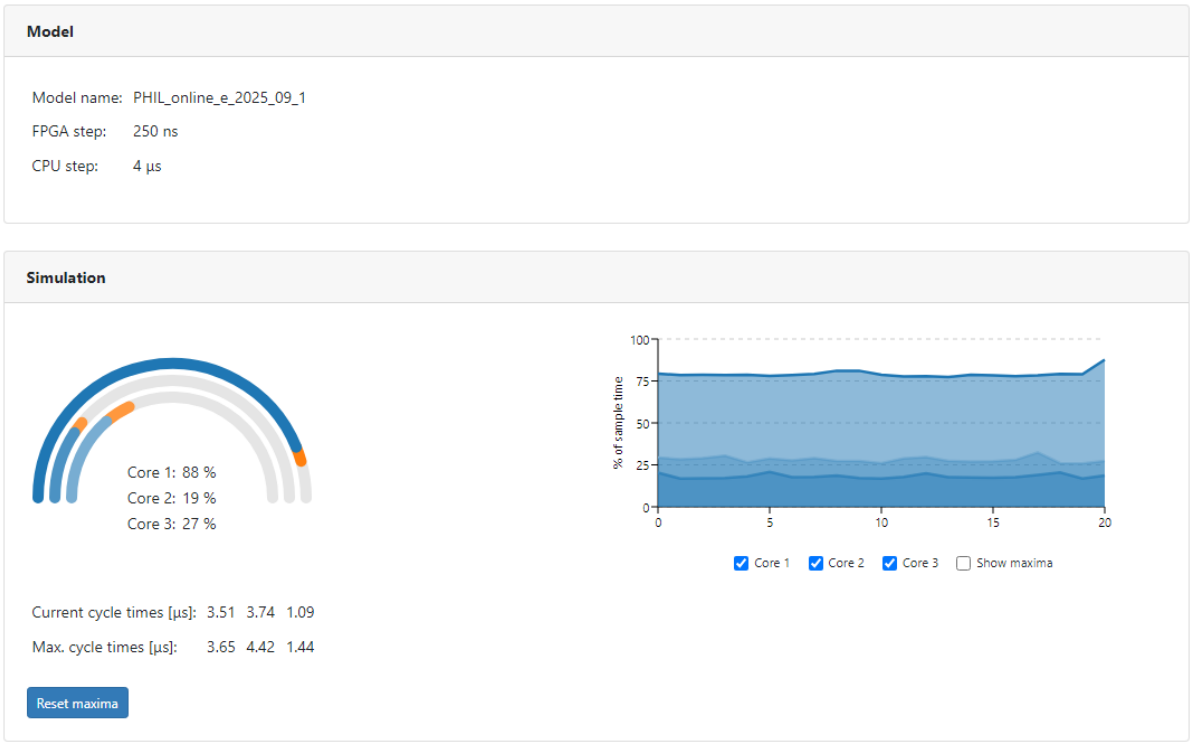


Figura 104:onere computazionale modello matematico

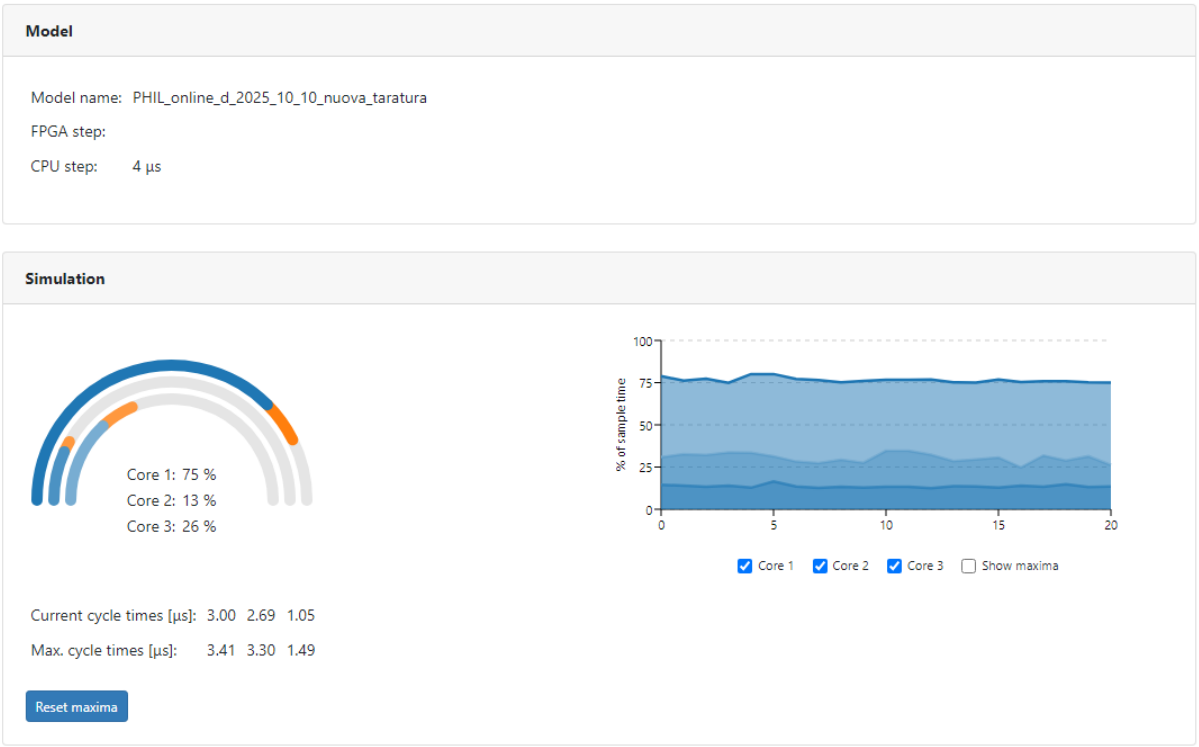


Figura 105: onere computazionale modello Plecs

Dalle immagini si osserva come entrambi i modelli presentino tempi di ciclo ampiamente inferiori al limite imposto dallo step di CPU (4 µs), garantendo quindi una stabilità e affidabilità dell'esecuzione in real-time.

Nel primo caso (modello matematico), l'onere computazionale risulta leggermente più elevato, con un utilizzo del Core 1 pari all'88 %, mentre nel secondo caso (fig. 105) il carico massimo scende al 75 %. Questo indica che la nuova implementazione risulta più ottimizzata dal punto di vista della distribuzione dei calcoli tra i core, riducendo il tempo medio di elaborazione per ciclo (da circa 3.5 µs a 3.0 µs). Il bilanciamento del carico tra i tre core è migliorato: si nota una diminuzione dell'impegno del Core 1, a fronte di una leggera variazione nei Core 2 e 3, mantenendo comunque un margine di sicurezza rispetto al tempo di campionamento.

Questo comportamento suggerisce che il modello Plecs, pur mantenendo una fedeltà di simulazione equivalente a quella del modello matematico, richiede minori risorse computazionali, rendendolo più adatto per applicazioni in Hardware-in-the-Loop (HIL) o in contesti dove è necessaria un'esecuzione a frequenze di campionamento più elevate.

In sintesi, l'analisi dell'onere computazionale conferma non solo la correttezza funzionale dei modelli, ma anche la superiorità in efficienza del blocco Plecs rispetto al modello matematico.

Modello LUT

L'implementazione del modello LUT nell'ambiente PHIL ha richiesto la modifica di alcune parti del file di azionamento in quanto è una macchina differenze rispetto a quella a magneti superficiali.

Inizializzazione

```
Init_encoder
entry: EnableCurrentRegulators = 1;
 EnableSpeedRegulator = 0;
 Counter = 0;
 Enable_DRV = 1;
 Timer=0;
 Mode_index=1;
during: id_ref_out = 0;
 iq_ref_out = 4;
 Counter = Counter +1;
 Timer = Counter * T_cycle;
 EncoderOffset = Encoder_counter
exit:
  %%EncoderOffset = EncoderOffset-pi/2;
```

Figura 106: modifica inizializzazione encoder

A differenza dei modelli lineari in cui l'inizializzazione viene effettuata iniettando una corrente di asse d, in questo modello abbiamo bisogno di iniettare una corrente di asse q (4 A). Questo perché la macchina a riluttanza pura non dispone di magneti e quindi il flusso non è allineato all'asse d come nel caso della macchina a magneti superficiali. Quindi iniettando una corrente di asse q, l'induttanza cambia in modo marcato quando il rotore si muove, questo crea una variazione periodica della corrente e del flusso, da cui si può estrarre la posizione del rotore.

Controllo di velocità

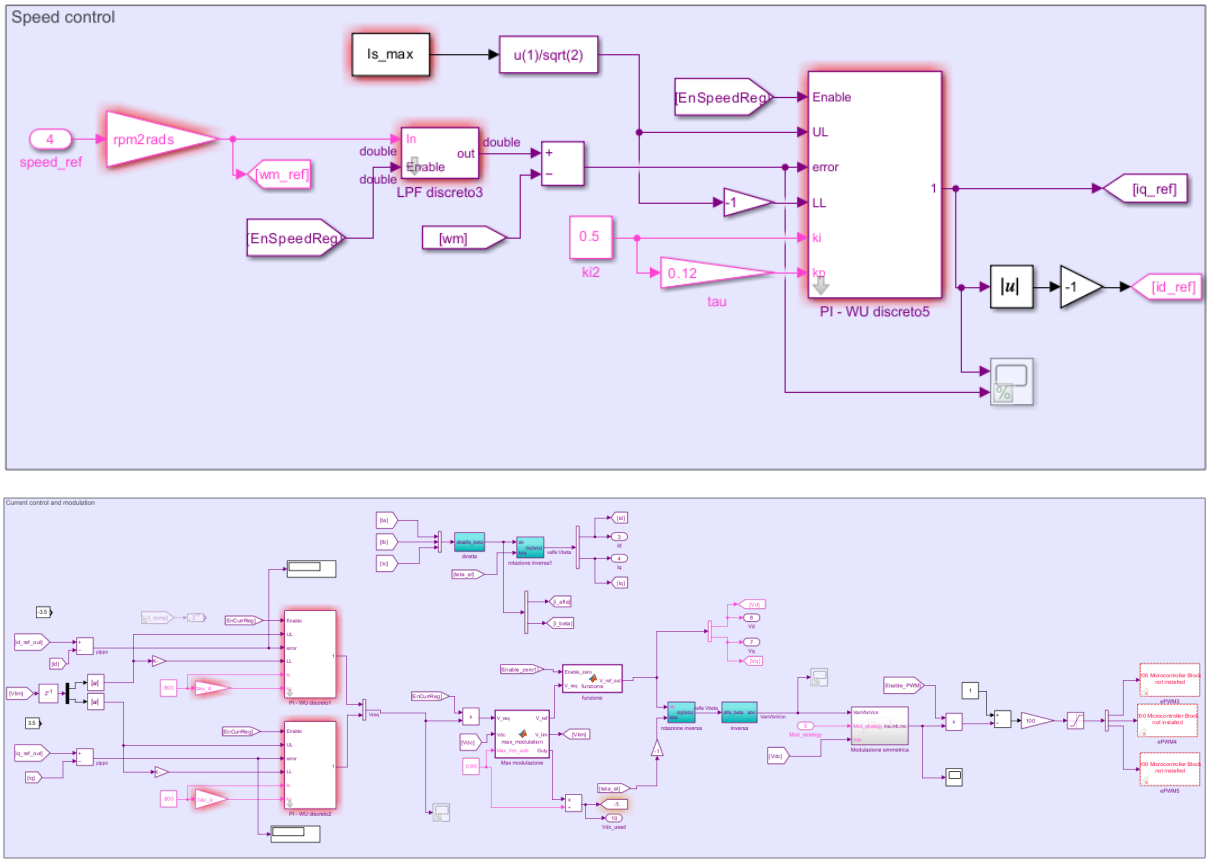


Figura 107: controllo di velocità modello LUT

A differenza del controllo della macchina PMSM, nel modello LUT le compensazioni delle FEM devono essere cancellate. In più dobbiamo ricordare che la coppia in questo caso dipenderà sia dalla corrente di asse q (come la PMSM) ma anche dalla corrente di asse d che dovrà quindi essere controllata.

Modificate queste differenze tra i modelli lineari e non è possibile implementare su PLECS il modello LUT:

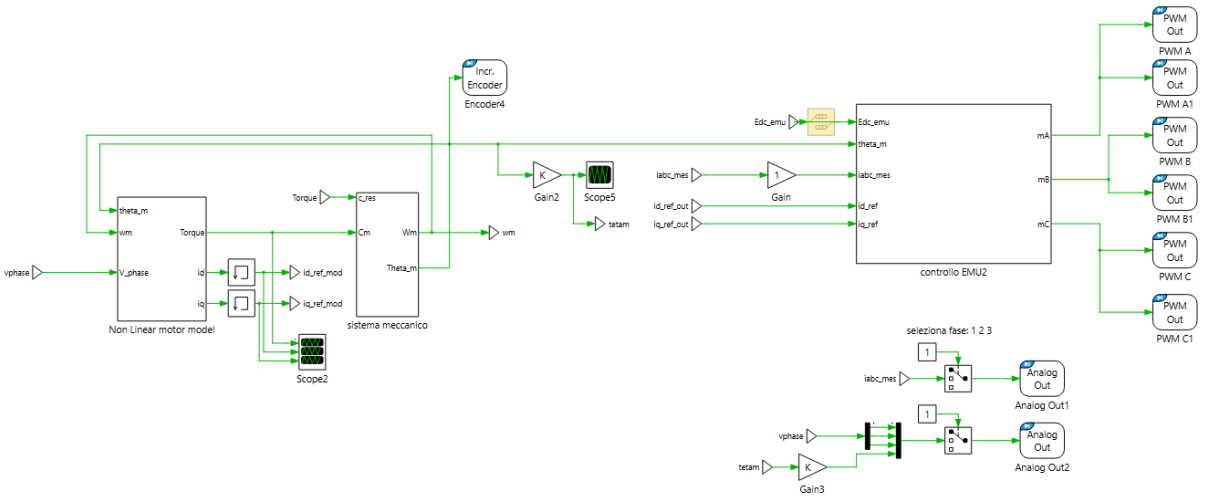


Figura 108: implementazione modello LUT non lineare su PLECS in PHIL

PHIL SIMULAZIONI

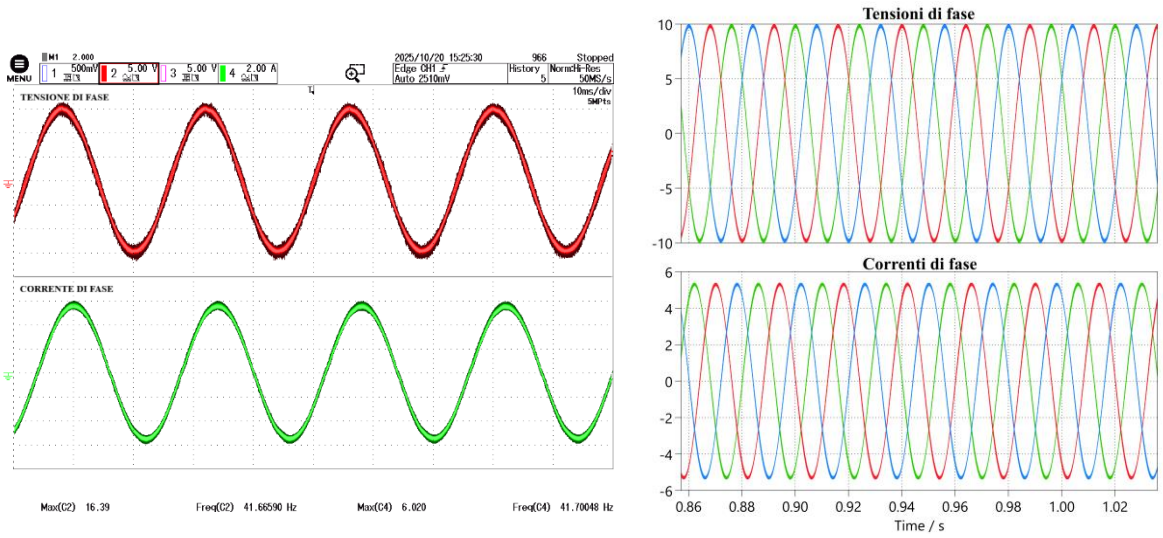


Figura 109: confronto tensione e corrente di fase a regime, modello LUT – 2500 rpm

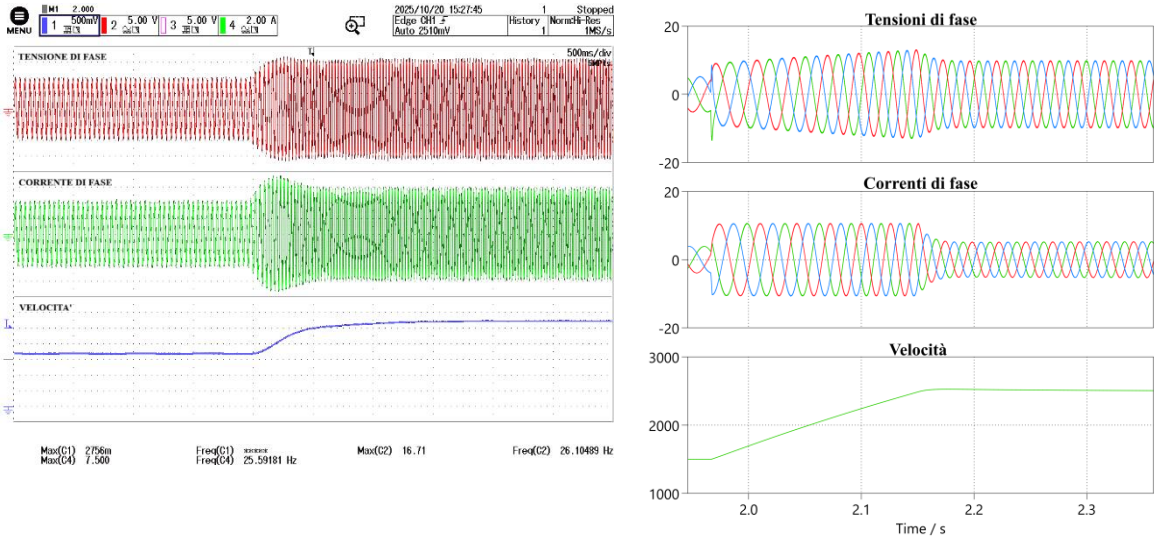
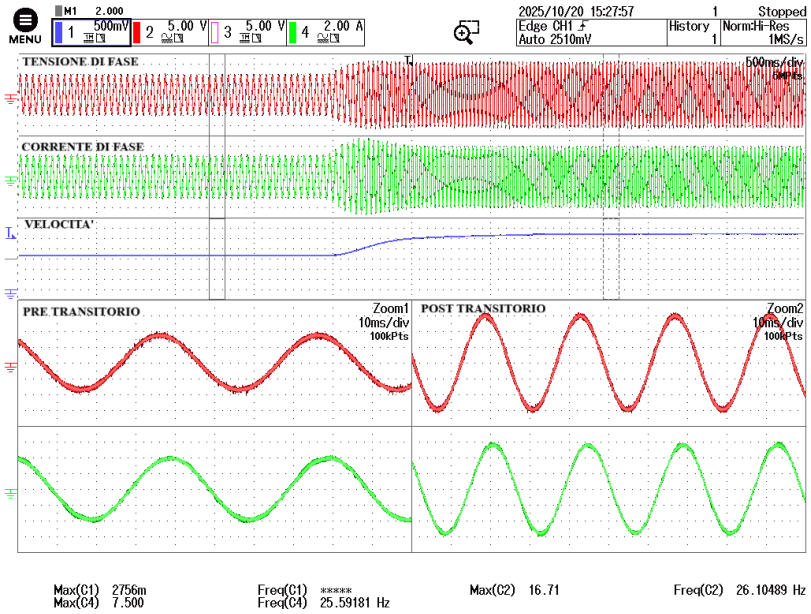
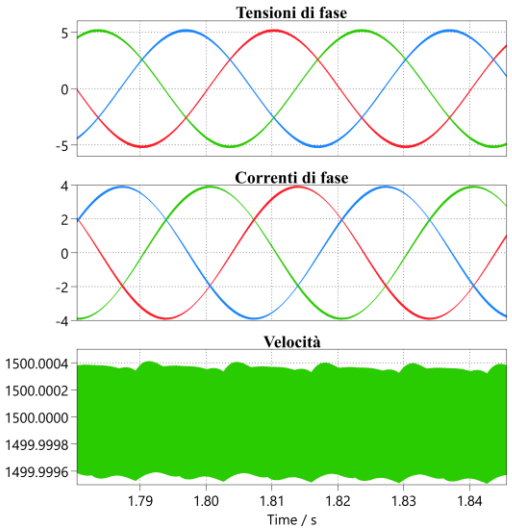


Figura 110: transitori di tensione, corrente e velocità (10mV/rpm), 1500-2500 rpm – modello LUT

PHIL



SIMULAZIONE PRE TRANSITORIO



SIMULAZIONI POST TRANSITORIO

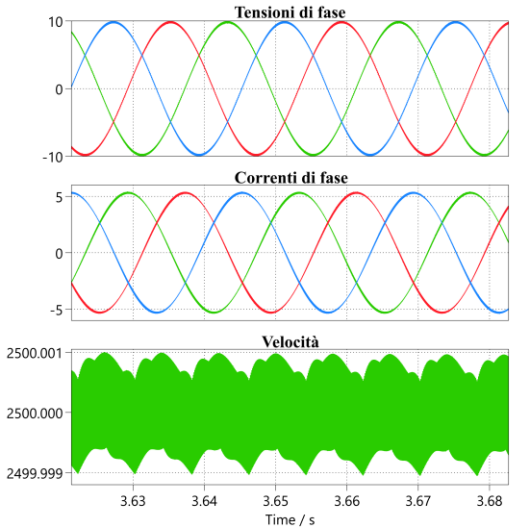


Figura 111: zoom tensioni e correnti pre e post transitorio – modello LUT

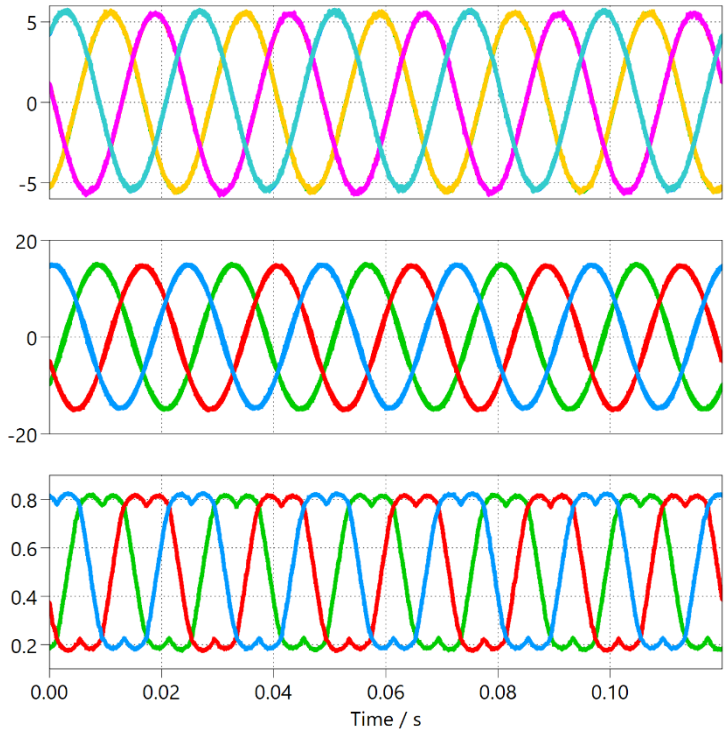


Figura 112: scope Plecs con correnti di fase e riferimenti, tensioni di fase e modulanti comando inverter – modello LUT

Modello flussi inversi

L'implementazione del modello a flussi inversi su PLECS viene fatta come per tutte le altre macchine tenendo conto delle modifiche fatte nel modello LUT, ovvero inizializzazione e controllo di velocità, che rimangono le stesse.

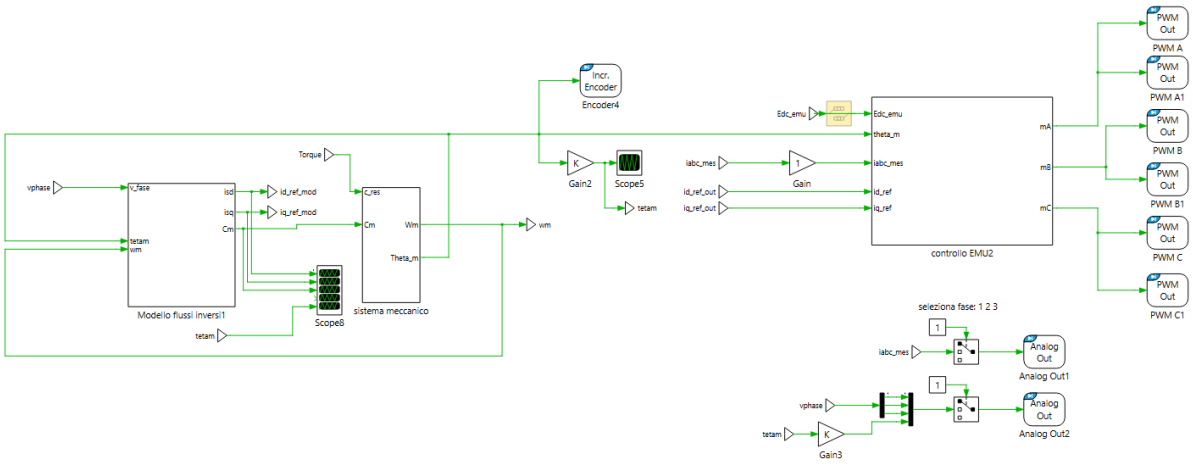


Figura 113: implementazione modello flussi inversi

PHIL SIMULAZIONI

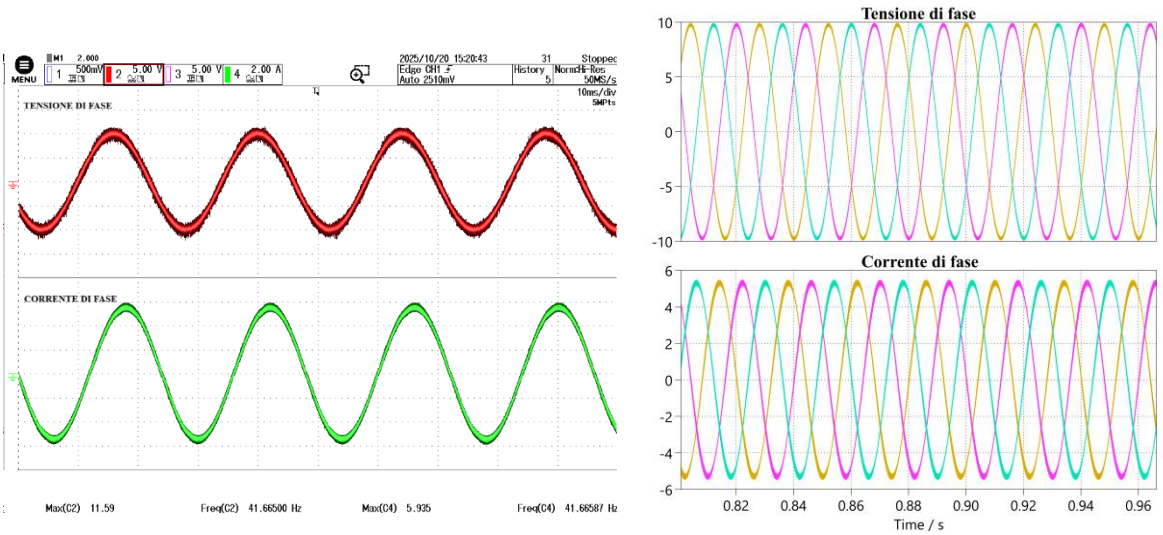


Figura 114:confronto tensione e corrente di fase a regime, modello flussi inversi – 2500 rpm

PHIL SIMULAZIONI

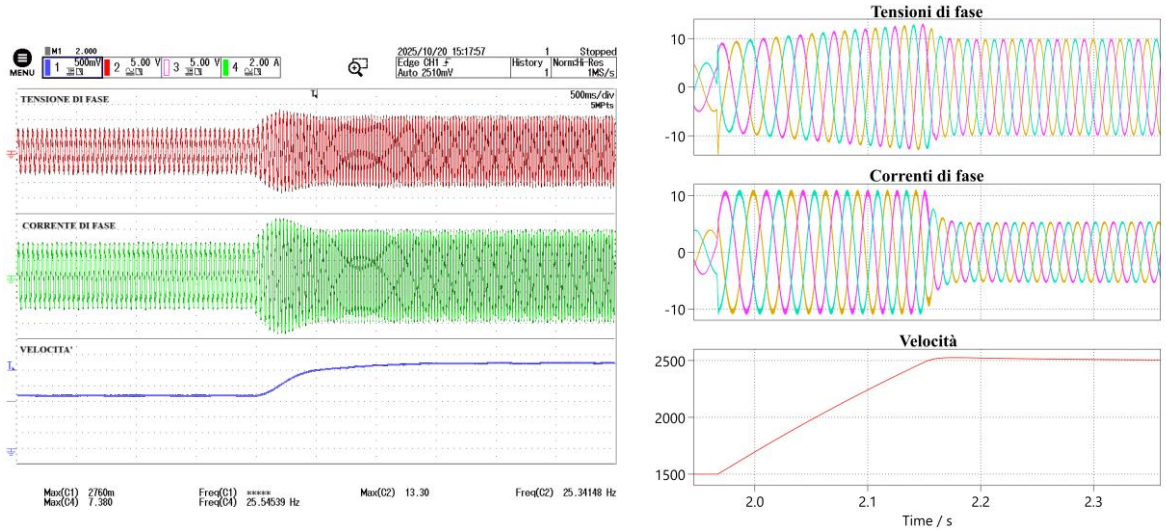
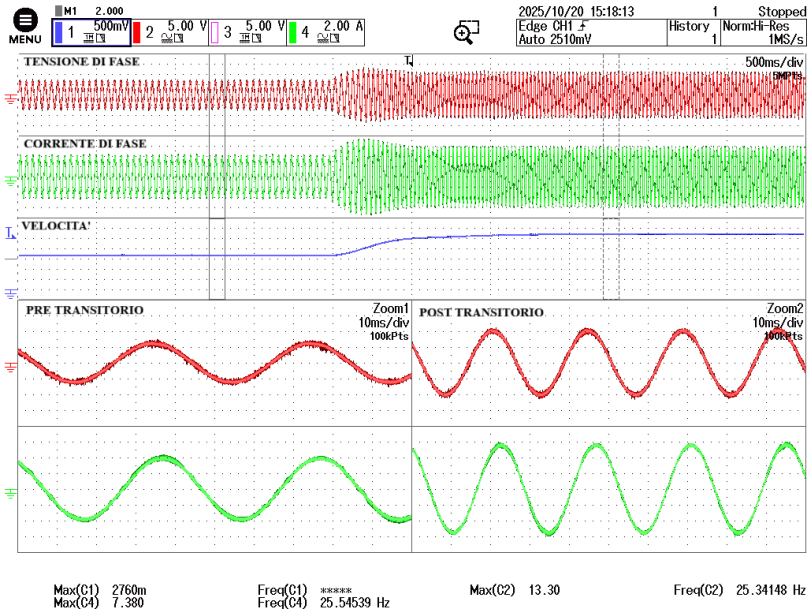
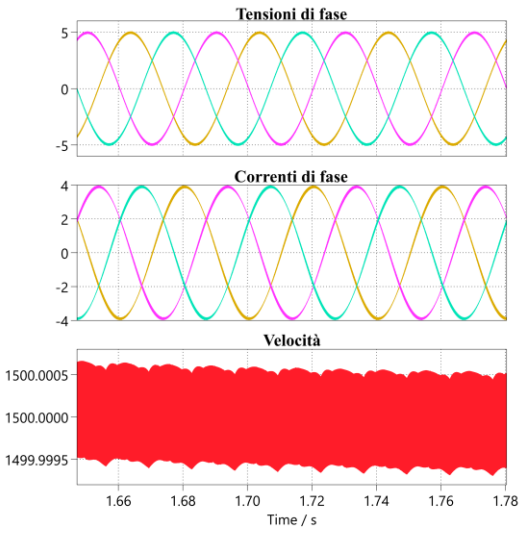


Figura 115: confronto transitori di corrente, tensione e velocità (10mV/rpm), modello flussi inversi – 0-2500 rpm

PHIL



SIMULAZIONI PRE TRANSITORIO



SIMULAZIONI POST TRANSITORIO

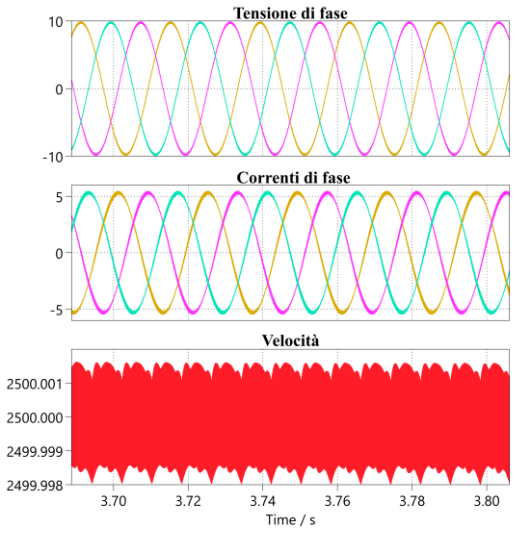


Figura 116: confronto zoom pre e post transitorio

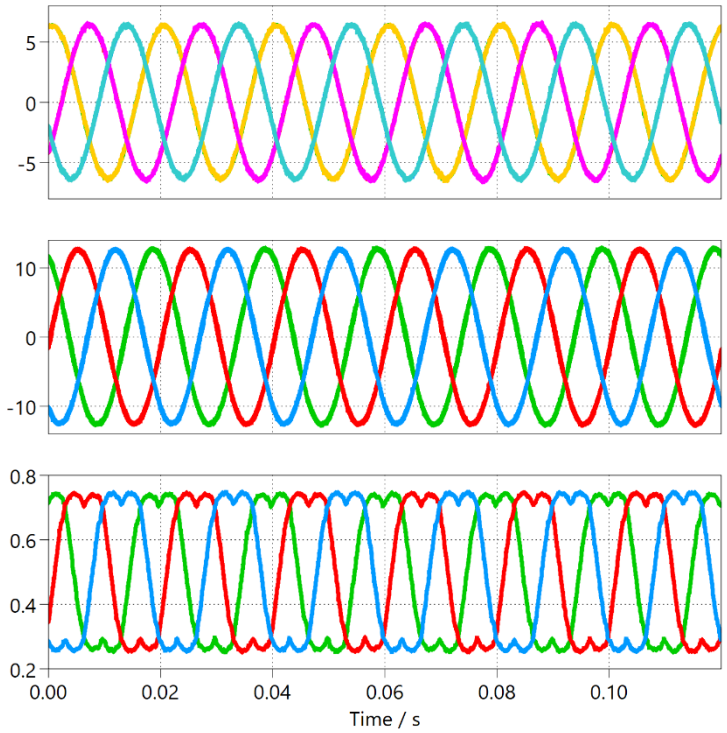


Figura 117:scope Plecs con correnti di fase e riferimenti, tensioni di fase e modulanti comando inverter – modello flussi inversi

Come già osservato precedentemente per i modelli lineari, anche i modelli non lineari risultano perfettamente equivalenti mostrando come la differenza tra simulazioni e PHIL sia davvero minima. Si può quindi affermare che l'architettura PHIL risulti affidabile quanto le simulazioni di qualsiasi software utilizzabile per azionamenti elettrici, con l'aggiunta della possibilità di poter testare il comportamento reale della macchina emulata.

Come ultimo confronto tra i modelli, come fatto per quelli lineari, è possibile osservare l'onere computazionale richiesto:

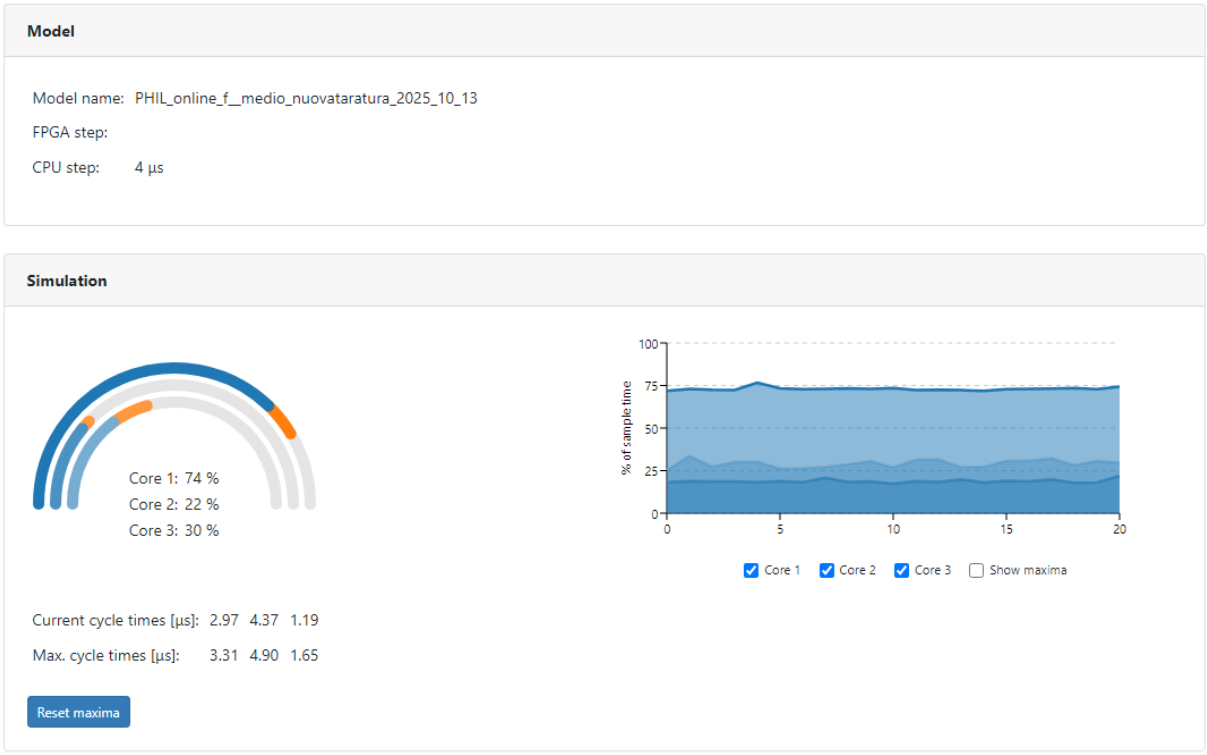


Figura 118: onere computazionale – modello LUT

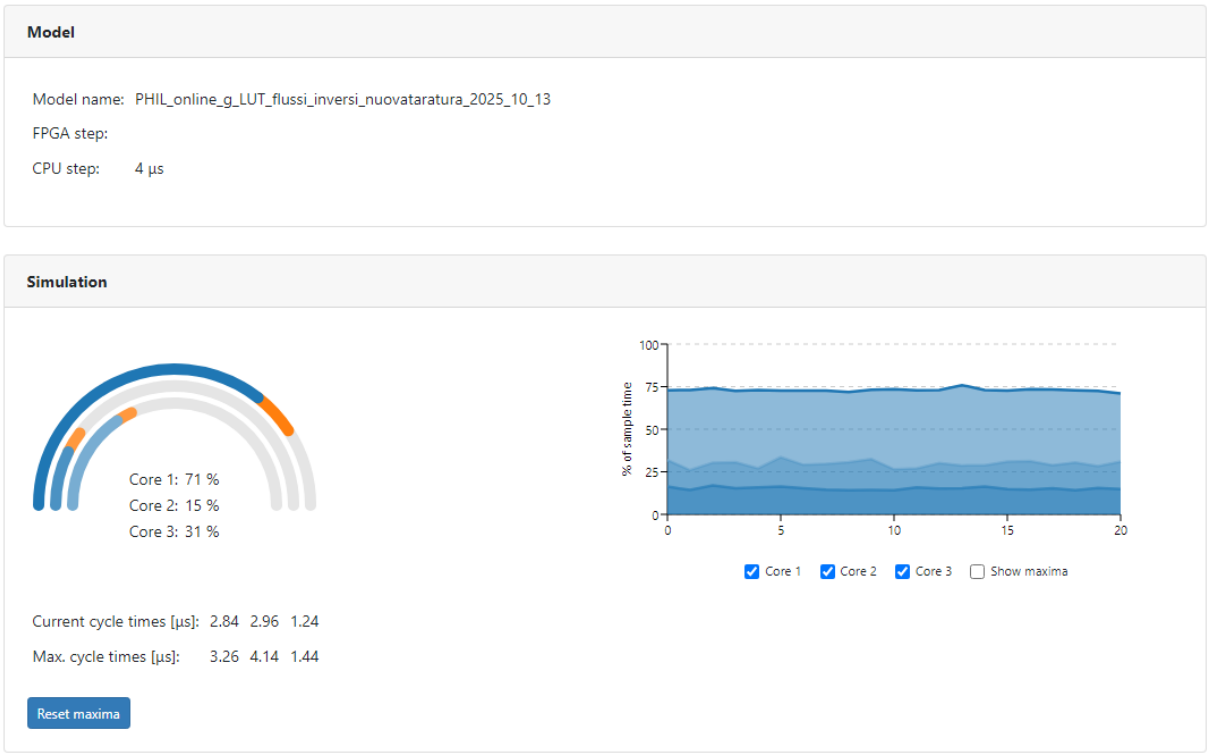


Figura 119: onere computazionale – modello flussi inversi

4.4 Flux mapping

Come si è potuto notare, per caratterizzare accuratamente una macchina reale è necessario conoscere la relazione non lineare tra le correnti i_d, i_qe i flussi concatenati φ_d, φ_q. Tali relazioni non possono essere ricavate in modo preciso da un semplice modello analitico, ma devono essere stimate sperimentalmente, costruendo delle mappe di flusso (flux maps) ottenute applicando correnti note alla macchina e misurandone le grandezze elettriche associate.

Un metodo comunemente utilizzato per questa identificazione è quello basato sulle equazioni della macchina a regime, sfruttando il fatto che, in condizioni stazionarie, le derivate dei flussi sono nulle. Partendo dalle equazioni del modello elettrico:

$$\begin{cases} v_{sd} = R_s i_{sd} + \dfrac{\partial \varphi_{sd}}{\partial t} - \omega \varphi_{sq} \\ v_{sq} = R_s i_{sq} + \dfrac{\partial \varphi_{sq}}{\partial t} + \omega \varphi_{sd} \end{cases}$$

e imponendo la condizione di regime ($\dot{\varphi}_d = \dot{\varphi}_q = 0$), si ottiene:

$$\begin{cases} v_{sd} = R_s i_{sd} - \omega \varphi_{sq} \\ v_{sq} = R_s i_{sq} + \omega \varphi_{sd} \end{cases}$$

Da queste espressioni i flussi possono essere ricavati direttamente:

$$\begin{cases} \varphi'_{sd} = \dfrac{v'_{sq} - R_s i'_{sq}}{\omega} \\ \varphi'_{sq} = -\dfrac{v'_{sd} - R_s i'_{sd}}{\omega} \end{cases}$$

Il pedice “primo” indica che tali misure appartengono alla prima prova di identificazione, eseguita in condizioni operative per cui $i_{sd} > 0$ e $i_{sq} > 0$, ossia nel secondo quadrante del piano $i_d - i_q$.

Compensazione della resistenza statorica

La difficoltà di questo metodo risiede nel fatto che le equazioni contengono la resistenza statorica R_s, la quale varia significativamente con la temperatura. Se non compensata, questa variazione introdurrebbe errori sistematici nelle mappe di flusso stimate.

Per eliminare la dipendenza da R_s, si esegue una seconda prova sperimentale consecutiva, speculare alla precedente, nella quale si applicano correnti tali da invertire la componente q:

$$\begin{cases} i''_{sd} = i'_{sd} \\ i''_{sq} = -i'_{sq} \end{cases}$$

Applicando le stesse equazioni di regime nella seconda prova si ottiene:

$$\varphi''_{sd} = \frac{v''_{sq} + R_s i'_{sq}}{\omega}$$
$$\varphi''_{sq} = -\frac{v''_{sd} - R_s i'_{sd}}{\omega}$$

A questo punto, combinando opportunamente i due risultati si ottengono espressioni indipendenti da R_s:

$$\tilde{\varphi}_{sd} = \frac{\varphi'_{sd} + \varphi''_{sd}}{2} = \frac{v'_{sq} + v''_{sq}}{2\omega}$$
$$\tilde{\varphi}_{sq} = \frac{\varphi'_{sq} - \varphi''_{sq}}{2} = -\frac{v'_{sd} + v''_{sd}}{2\omega}$$

La media elimina completamente il contributo resistivo e consente di ricavare i flussi reali legati alla macchina, indipendentemente dalle variazioni termiche.

4.4.1 Risultati flux mapping

L'implementazione del flux mapping, è stato realizzato attraverso un codice Matlab inserito all'interno di una function in Simulink. Il software di controllo del DUT è rimasto praticamente invariato se non per l'aggiunta di questa parte che può essere anche un ottimo elemento di verifica per capire se il funzionamento dei modelli non lineari risponde correttamente in entrambi i casi.

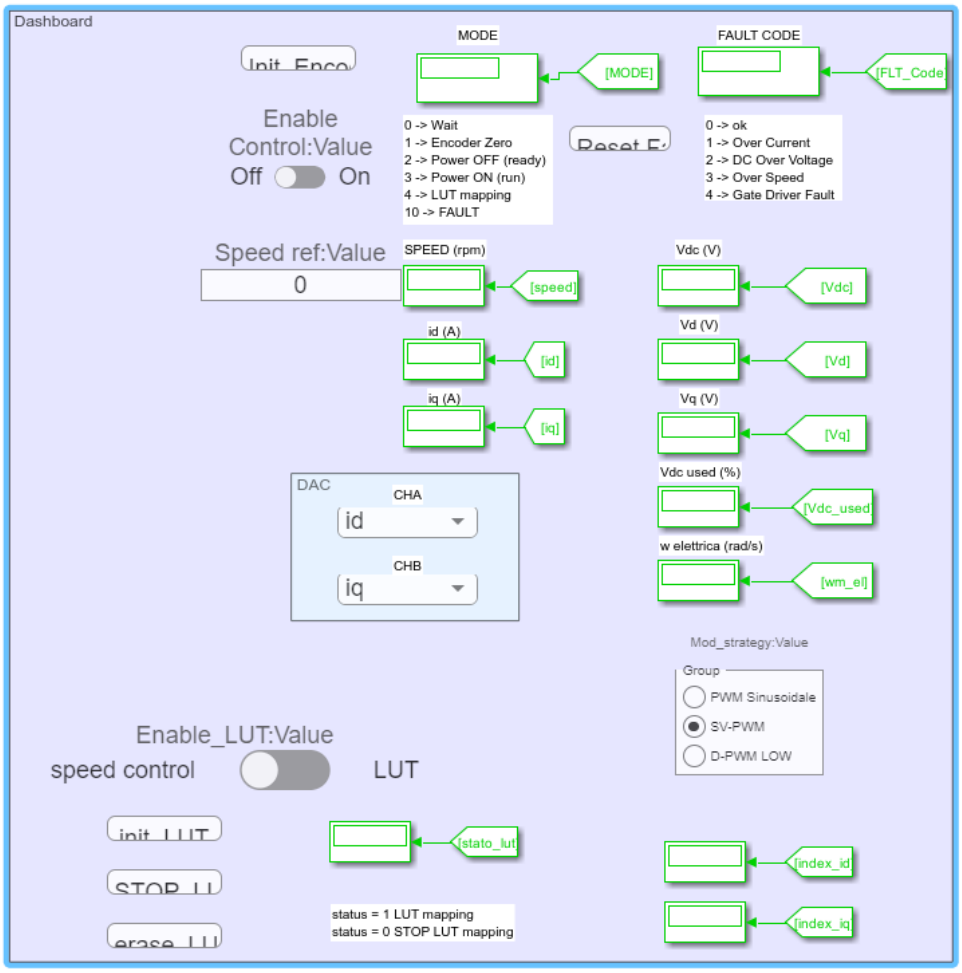


Figura 120: dashboard con implementazione flux mapping

Come si può vedere la dashboard di controllo e monitoraggio della DUT è stata modificata aggiungendo uno switch che ci permetterà di scegliere come far operare il controllo.

Fintanto che lo switch sarà su speed control, il valore associato alla variabile "Enable_LUT" sarà pari a 0 e quindi avremo il classico funzionamento visto in precedenza in cui il tutto sarà regolato dal controllo di velocità.

Nel momento in cui si sceglie di switchare su LUT la variabile "Enable_LUT" assumerà valore pari ad 1 e si entrerà in nuovo algoritmo di generazione delle mappe di flusso comandato dalla state machine (vedi figura seguente).

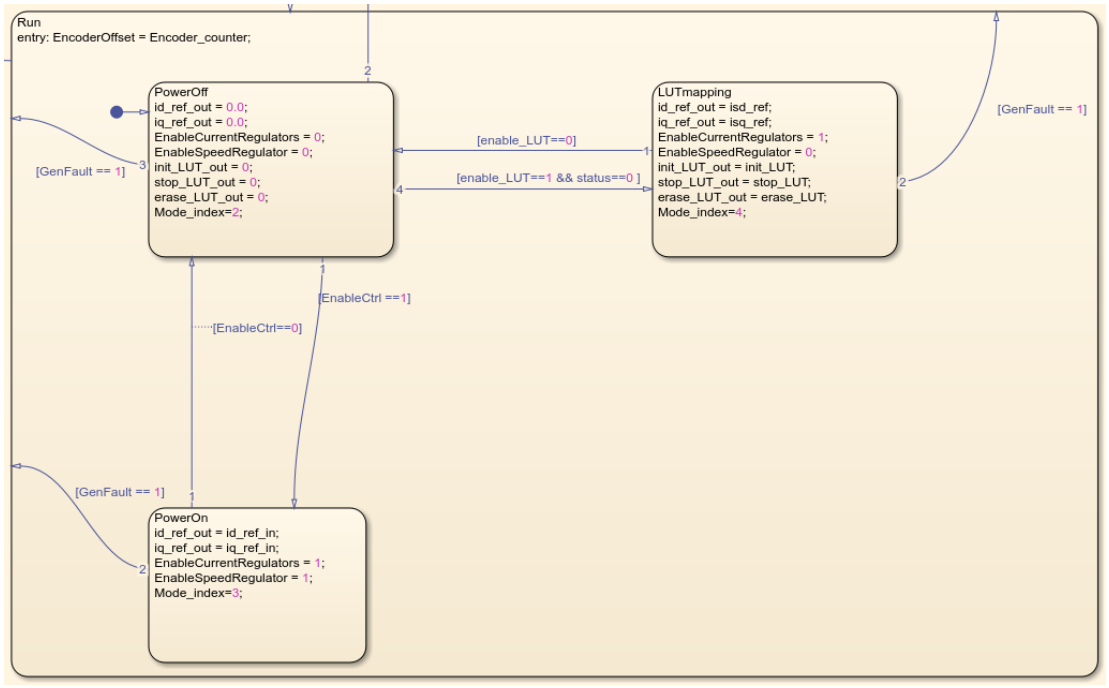


Figura 121: state machine con aggiunta del flux mapping

A questo punto il software sarà pronto per acquisire i valori di flusso al variare delle correnti; per farlo basterà cliccare sulla dashboard il pulsante “init_LUT”.

In questo modo inizierà il vero e proprio flux mapping grazie al seguente algoritmo

Azionamento_riluttanza_flux_mapping_v20x2820x29 ▸ Triggered Subsystem ▸ flux_mapping 3

```
function [isd_ref, isq_ref, isd_vector_out, isq_vector_out, cnt, Flux_d_out, Flux_q_out,...
    Flux_d_conj_out, Flux_q_conj_out, Flux_map_d_out, Flux_map_q_out, V_map_d_out, V_map_q_out,...
    V_map_d_conj_out, V_map_q_conj_out, mode, k_id_out, k_iq_out] = flux_map(start_sequence,...
    stop_sequence, erase_maps, isd_filt, isq_filt, vsd_filt, vsq_filt, w_filt, Rs, isd_ref_OL, isq_ref_OL, set_time, Tc)

persistent status index_isd index_isq timer isd_vector isq_vector dim_isd dim_isq flag_conj Flux_d Flux_q V_map_d...
    V_map_q Flux_d_conj Flux_q_conj V_map_d_conj V_map_q_conj Flux_map_d Flux_map_q;

if isempty(index_isd)
    % Vettori di riferimento
    isd_vector =linspace(0.5,5,7)*(-1);   % solo negativi per asse d
    isq_vector = linspace(0.5,5,7);         % solo positivi per asse q

    dim_isd = length(isd_vector);
    dim_isq = length(isq_vector);

    index_isd = 1;
    index_isq = 1;

    timer = 0;
    status = 0;
    %M1=ones(4);
    %M2=ones(4);

    Flux_d = ones(7);
    Flux_q = ones(7);
    Flux_d_conj = ones(7);
    Flux_q_conj = ones(7);
    Flux_map_d = ones(7);
    Flux_map_q = ones(7);
    V_map_d = ones(7);
    V_map_q = ones(7);
    V_map_d_conj = ones(7);
    V_map_q_conj = ones(7);
```

Figura 122: codice flux mapping parte 1

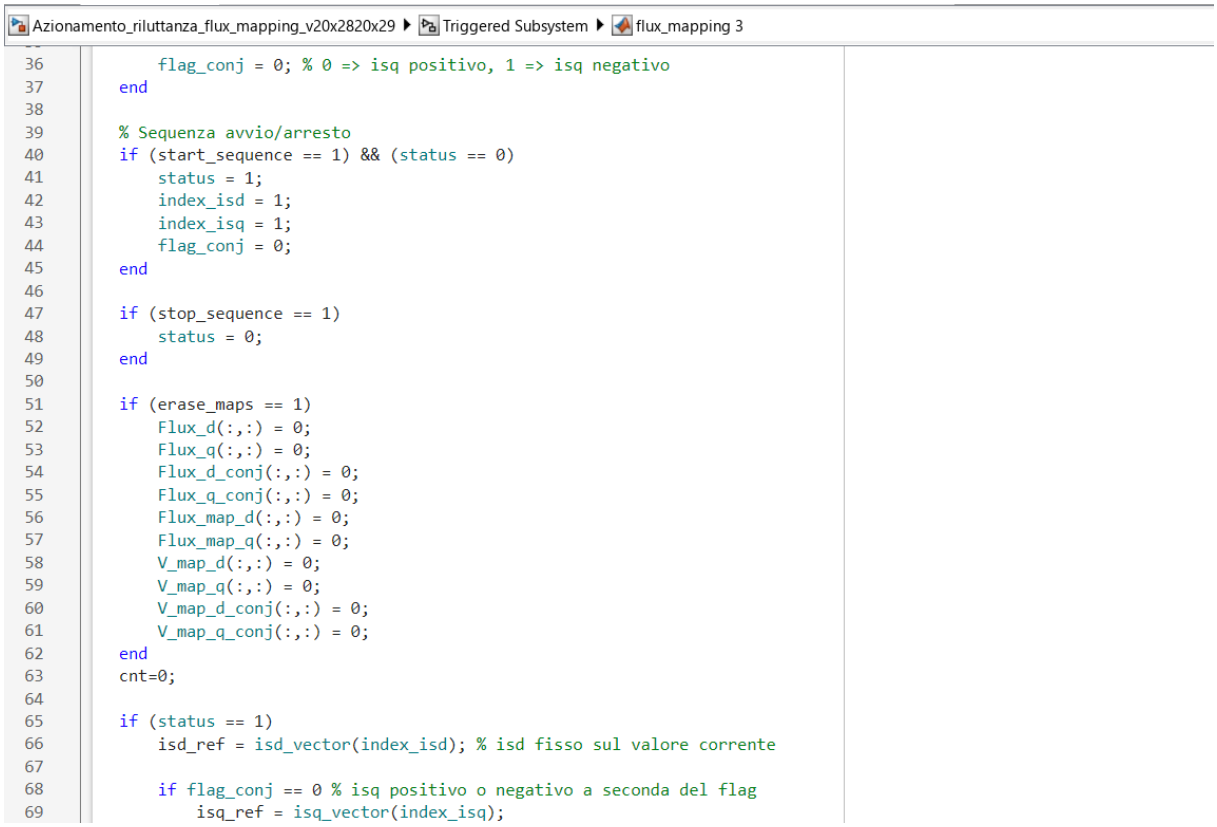


Figura 123: codice flux mapping parte 2

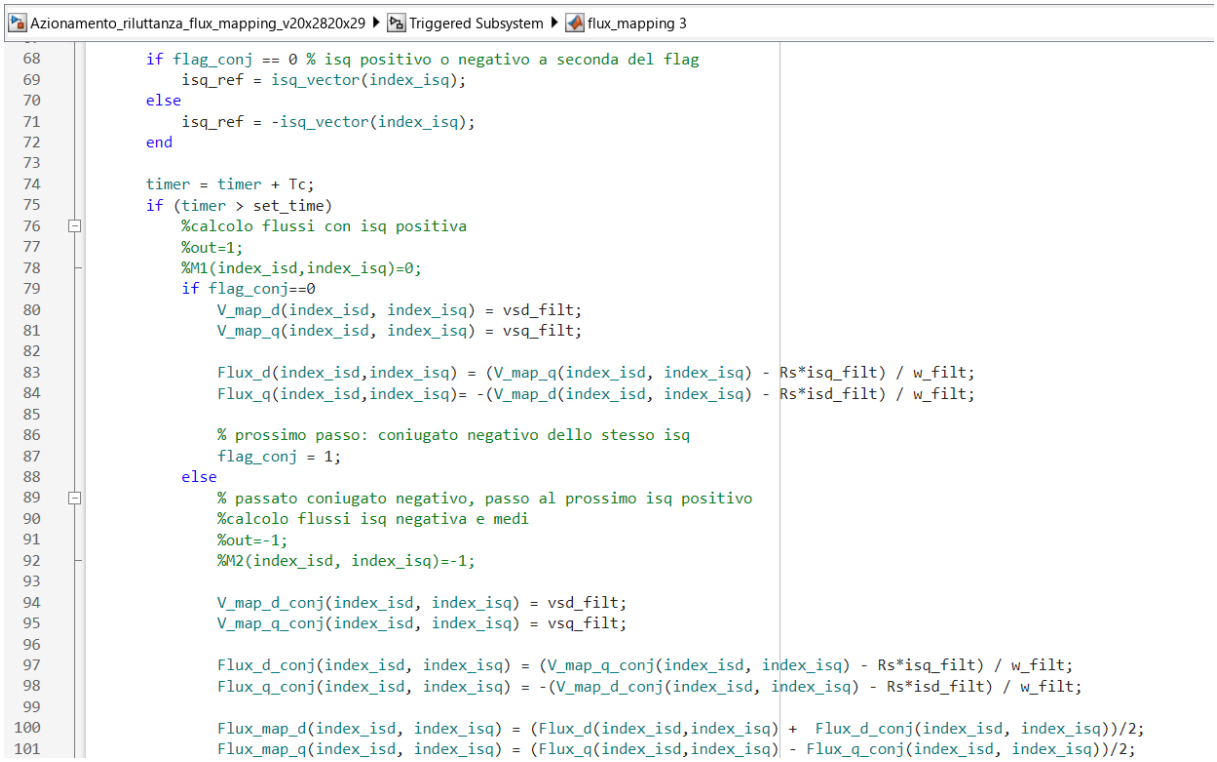


Figura 124: codice flux mapping parte 3

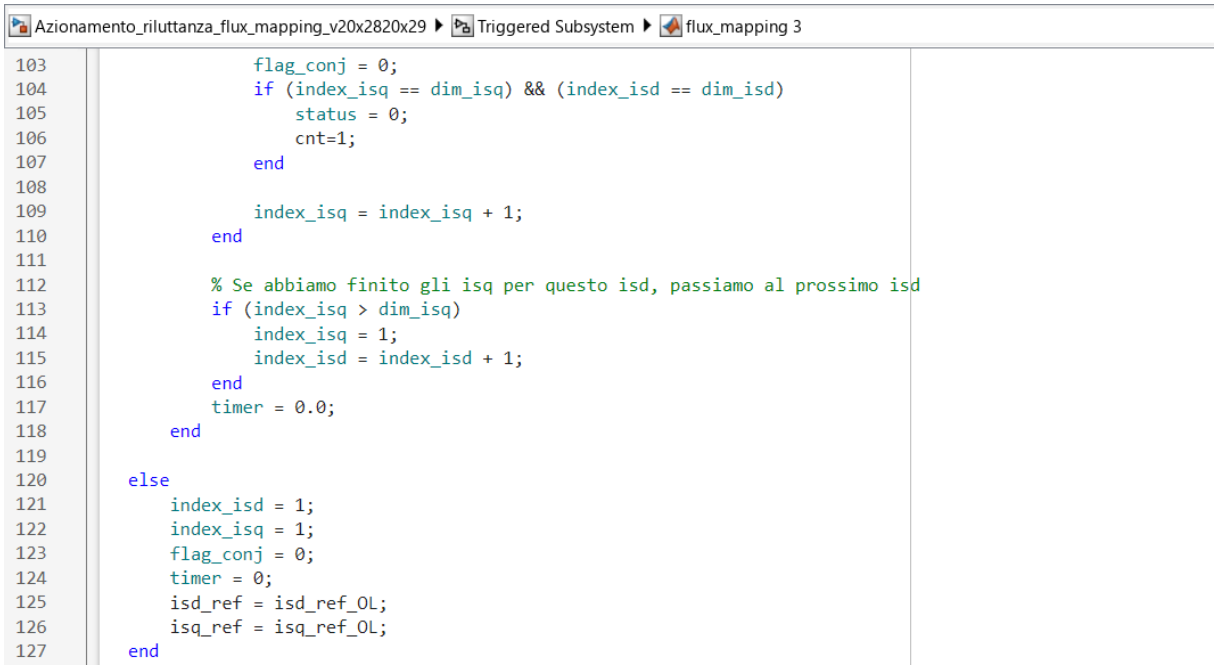


Figura 125: codice flux mapping parte 4

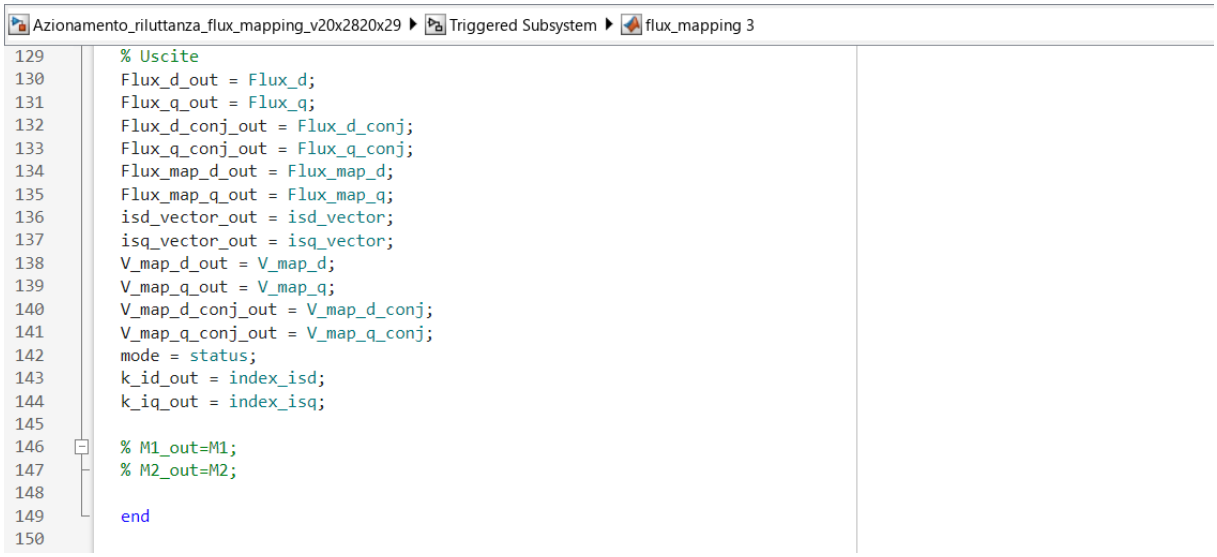


Figura 126: codice flux mapping parte 5

L'algoritmo implementato permette di generare sperimentalmente la mappa dei flussi magnetici della macchina nel piano delle correnti (i_d, i_q), necessaria per la caratterizzazione non lineare del modello. A differenza dello sviluppo teorico fatto nel capitolo 2, questa procedura definisce come le grandezze vengono effettivamente misurate, acquisite e organizzate per popolare le matrici di flusso e di tensione utilizzate nel modello PHIL.

L'algoritmo opera secondo una logica automatizzata, basata sull'iniezione sequenziale di coppie di correnti (i_d, i_q). Vengono inizialmente generati due vettori di riferimento: un insieme di valori negativi per l'asse d e un insieme di valori positivi per l'asse q. La griglia risultante rappresenta tutti i punti del piano (i_d, i_q)sui quali si desidera ricostruire i flussi magnetici. Il sistema, una volta avviata la procedura,

si posiziona su ciascun punto della griglia imponendo i riferimenti di corrente corrispondenti; un timer interno garantisce che, prima della misura, il sistema abbia raggiunto uno stato stazionario.

Figura 127: immagine oscilloscopio dell'iniezione delle correnti per la mappatura

Per ogni combinazione (i_d, i_q), l'algoritmo acquisisce le tensioni filtrate v_de v_q, le correnti effettive e la velocità elettrica filtrata. Tali grandezze vengono utilizzate per ricostruire i flussi φ_d e φ_q mediante le equazioni inverse del modello elettrico.

Un aspetto fondamentale dell'algoritmo è costituito dalla doppia misura rispetto all'asse q. Per ciascun valore positivo di i_q, la stessa misura viene ripetuta imponendo il valore opposto $-i_q$. Questa operazione permette di eliminare gli effetti simmetrici e parasimmetrici derivanti dalla saturazione magnetica e di ottenere una stima più accurata delle componenti dispari del flusso, come descritto nella teoria precedente. Le mappe finali di flusso vengono quindi calcolate mediando opportunamente le misure con $+i_q$ e $-i_q$, generando così matrici più regolari e coerenti con il comportamento fisico della macchina.

Al termine della procedura, quando tutti i valori dei due vettori di riferimento sono stati esplorati, l'algoritmo segnala il completamento della mappatura e restituisce in uscita tutte le matrici ricostruite: flussi primari Φ_de Φ_q, flussi coniugati, mappe finali filtrate e le corrispondenti mappe delle tensioni. Questi risultati costituiscono la base dati utilizzata dal modello non lineare sviluppato nei capitoli precedenti e permettono di confrontare i risultati ottenuti in simulazione con quelli misurati nella configurazione Power Hardware-in-the-Loop.

Si è cercato quindi di confrontare le mappe di flusso ottenute inizialmente tramite le derivate parziali con quelle ottenute tramite flux mapping, ne segue il risultato.

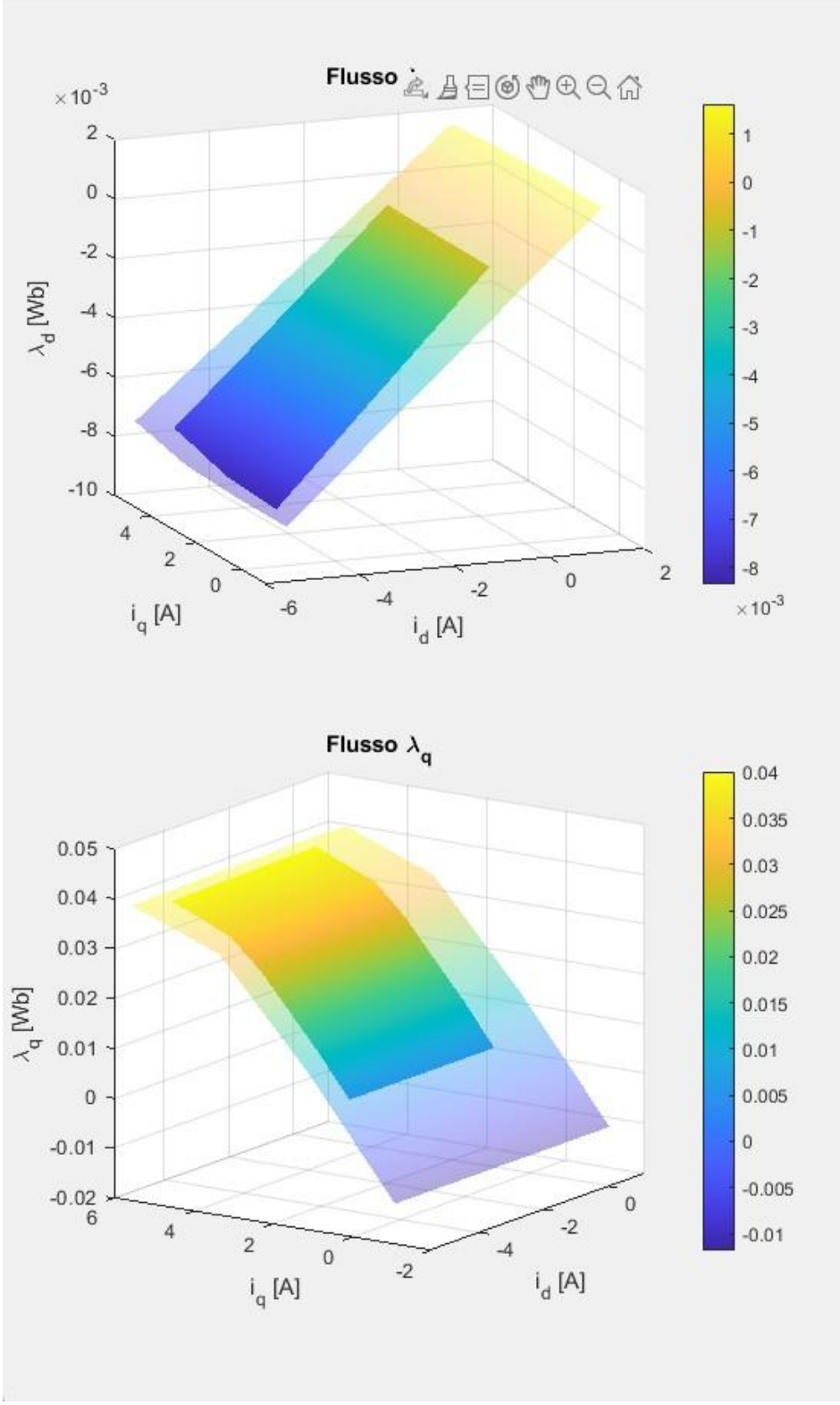


Figura 128: confronto mappe di flusso generate con derivate parziali e flux mapping

Ne consegue quindi che il funzionamento in PHIL può essere implementato anche con questa tipologia di generazione delle mappe di flusso nel caso la resistenza R_s porti non linearità non desiderate.

Capitolo 5

Conclusioni

Il lavoro svolto in questa tesi ha consentito di sviluppare, analizzare e validare un insieme di modelli lineari e non lineari rappresentativi del comportamento dinamico del sistema oggetto di studio. Nei capitoli iniziali, tali modelli sono stati realizzati e testati mediante simulazioni in ambiente PLECS, verificandone con accuratezza la coerenza fisica, la stabilità e la sensibilità ai parametri principali. Successivamente, gli stessi modelli sono stati implementati in un'architettura Power Hardware-in-the-Loop (PHIL), al fine di valutarne il comportamento in condizioni sperimentali realistiche e in presenza di un dispositivo fisico sotto test.

L'integrazione dei modelli nel banco PHIL ha richiesto una fase di adattamento e ottimizzazione legata all'esecuzione real-time, alla gestione dei segnali di interfaccia e alla definizione dell'interfaccia di potenza. Nonostante tali criticità, l'intero sistema ha mostrato stabilità e riproducibilità adeguate, consentendo di effettuare un confronto diretto tra i risultati della simulazione pura e quelli ottenuti in ambiente hardware.

L'analisi sperimentale ha evidenziato una notevole corrispondenza tra i risultati PHIL e quelli ottenuti tramite simulazione PLECS, sia nel caso dei modelli lineari sia per quelli non lineari. Le grandezze elettriche principali (tensioni, correnti, fasi d-q, variabili di stato) hanno mostrato scostamenti limitati e compatibili con gli effetti attesi dell'hardware reale, confermando l'accuratezza del modello e la validità dell'interfaccia PHIL adottata. In particolare, la capacità del sistema di riprodurre le dinamiche non lineari dimostra la robustezza dell'approccio impiegato e la sua idoneità a supportare futuri studi sperimentali e applicativi.

In conclusione, l'attività svolta ha permesso di dimostrare che i modelli implementati in architettura Power Hardware-in-the-Loop replicano con buona affidabilità i risultati ottenuti in simulazione, consolidando l'efficacia del metodo PHIL come strumento di validazione intermedia tra simulazione numerica e sperimentazione su impianto reale. Le metodologie e i risultati ottenuti rappresentano inoltre un punto di partenza per ulteriori sviluppi, quali l'estensione dell'approccio a modelli più complessi, l'ottimizzazione delle strategie di controllo, l'analisi dei fenomeni di instabilità PHIL e l'integrazione con sistemi distribuiti o rinnovabili su larga scala.

Bibliografia

[1] Plexim, «Supporto Web Plexim,» Plexim, 2018. [Online]. Available: www.plexim.com.

[2] P. M. Mengoni, «Azionamenti Elettrici per Applicazioni Industriali ed Eoliche M,» Slide del corso, Bologna, 2024.

[3] C. G. A. F. L. Z. A. C. Yulong Cui, «High-Speed Synchronous Reluctance Motors with Additively Manufactured Rotors,» *IEEE Energy Conversion Congress and Exposition (ECCE),* pp. 979-8-3503-7606-7/24, 2024.

[4] G. D. P. R. a. G. P. Simone Ferrari, «The dq-theta Flux Map Model of Synchronous Machines,» *IEEE Energy Conversion Congress and Exposition (ECCE) tenutosi a Vancouver, BC, Canada nel 10-14 Oct. 2021,* pp. 3716-3723, 2021.

[5] L. Vita, *Sviluppo di un sistema Hardware In the Loop per l'emulazione di macchine elettriche,* Bologna, 2022/2023.