UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Studio e sviluppo di un algoritmo di Controllo Predittivo basato su Modello per azionamenti elettrici

Design and development of a Model Predictive Control algorithm for electric drives

Tesi di Laurea di:
Andrea Iannotti

Relatore:
Prof. Gianluca Ippoliti

Correlatore:
Prof. Giuseppe Orlando

Anno Accademico 2025/2026

Indice

1 Introduzione **1**

1.1 Architettura Funzionale 1

1.1.1 Convertitore di Potenza 2

1.1.2 Motore Elettrico 3

1.1.3 Trasduttori e Sensori 4

1.1.4 Sistema di Controllo 5

1.2 Obiettivi della tesi 7

1.2.1 Struttura della tesi 7

2 Modellazione del Motore Sincrono a Magneti Permanenti (PMSM) **9**

2.1 Trasformata di Park 10

2.1.1 Modello Bifase Equivalente 10

2.1.2 Passaggio al sistema di riferimento ruotante 11

2.1.3 Vantaggi della trasformazione 11

2.2 Modellazione del Motore Sincrono a Magneti Permanenti (PMSM) 12

2.2.1 Relazioni elettromagnetiche e transizione al sistema rotante 13

2.2.2 Equazione della coppia e analogia con la macchina a collettore . . . 15

2.3 Azionamenti con motore AC brushless 15

2.4 Campo di operatività 16

2.5 Necessità di una strategia di controllo avanzata 17

3 Algoritmo di Controllo (Model Predictive Control) **19**

3.1 Modellazione discreta del sistema 20

3.1.1 Formulazione del modello a stato esteso 21

3.2 Fondamenti del Controllo Predittivo (MPC) 22

3.3 Formulazione del funzionale di costo 25

3.4 Gestione dei vincoli operativi 26

3.4.1 Formulazione del problema QP 26

3.4.2 Ottimizzazione vincolata e dualità 27

3.4.3 L'algoritmo di ottimizzazione di Hildreth 28

3.5 Funzioni di Laguerre 29
3.5.1 Funzionale di costo parametrizzato 30
3.5.2 Minimizzazione del funzionale e connessione con il DLQR 31
3.5.3 Analisi della soluzione ottima 31
3.5.4 Legge di controllo a retroazione di stato 32
3.6 Stabilità e robustezza del sistema 33
3.6.1 Stabilità con vincoli sullo stato terminale 33
3.6.2 Stabilità con orizzonte di predizione esteso 34

4 Implementazione dell'algoritmo di controllo nell'ambiente di sviluppo 35
4.1 Validazione in Ambiente di Simulazione 36
4.2 Traduzione del modello teorico in ambiente di calcolo 37
4.2.1 Discretizzazione e costruzione del modello esteso 38
4.2.2 Selezione dei parametri di controllo e delle funzioni di Laguerre . . 38
4.2.3 Configurazione dinamica dei vincoli 40
4.2.4 Ottimizzazione offline e preparazione del solutore 41
4.2.5 Analisi di robustezza tramite Gain Scheduling 42
4.3 Integrazione nel modello Simulink: il solutore real-time 44
4.3.1 Analisi del codice di controllo real-time 45
4.4 Stima online della resistenza statorica 49
4.4.1 Stima dei parametri e osservatori di stato 50
4.4.2 Implementazione dello stimatore di resistenza 51

5 Validazione sperimentale e confronto con l'ambiente di simulazione 55
5.1 Configurazione del setup sperimentale 56
5.1.1 Interfaccia di comunicazione e controllo 57
5.2 Confronto Simulazione-Esperimento: Test Dinamici 58
5.2.1 Analisi della risposta dinamica in simulazione 58
5.2.2 Analisi della risposta dinamica su banco prova 61
5.3 Analisi dell'architettura con Estimatore SMO 63
5.4 Analisi critica delle non-idealità 65
5.4.1 Discrepanze tra modello e sistema reale 65
5.4.2 Validazione della robustezza 66
5.5 Analisi comparativa delle prestazioni 66
5.5.1 Interpretazione dei risultati 67

6 Conclusioni e sviluppi futuri 68
6.1 Sintesi dei risultati raggiunti 68
6.2 Analisi critica e sviluppi futuri 69

Elenco delle figure

1.1 Schema a blocchi di un azionamento elettrico 1
1.2 Convertitori statici 3
1.3 Motori elettrici 4
1.4 Sensori di corrente ad effetto Hall montati sull'inverter 5
1.5 Microcontrollore Texas Instruments 6

2.1 Proiezioni assi trifase su un sistema di riferimento ortogonale 11
2.2 Statore motore brushless 12
2.3 Rotore motore brushless 13
2.4 Campo di operatività dell'azionamento del motore AC brushless 17

4.1 Architettura gerarchica: anello esterno di velocità e anello interno basato su MPC. 36
4.2 Interfaccia del blocco MPC: ingressi, uscite e logica di retroazione. 44
4.3 Schema del blocco Simulink per lo stimatore di resistenza. 51

5.1 Setup sperimentale: PC di controllo, microcontrollore TI C2000 e motore PMSM Teknic. 56
5.2 Microcontrollore 57
5.3 Motore brushless 58
5.4 Dinamica della velocità 59
5.5 Evoluzione dell'indice di *Gain Scheduling* 60
5.6 Tracking della corrente di coppia i_q 61
5.7 Tracking della velocità 61
5.8 Risposta sperimentale a gradini di velocità e iniezione di carico tramite $i_{q,ref}$. 62
5.9 Simulazione dell'estrazione di $\hat{R}_s$ con campionamento a $5\,ms$ e correlazione con l'indice di *Gain Scheduling*. 64

Capitolo 1

Introduzione

Negli ultimi decenni, il settore dell'automazione industriale ha subito una trasformazione radicale, trainata dalla crescente richiesta di efficienza, precisione e dinamica operativa. Al centro di questa evoluzione si pongono gli azionamenti elettrici, componenti fondamentali non solo nell'automotive, ma anche nella robotica e nei processi di produzione avanzati. Questi dispositivi regolano la velocità, la direzione e la coppia di un motore elettrico, migliorando l'efficienza e la precisione delle macchine a cui sono applicati. L'utilizzo degli azionamenti elettrici si estende a vari settori, dall'industria manifatturiera ai sistemi di trasporto, fino alle applicazioni domestiche.

1.1 Architettura Funzionale

Un azionamento elettrico può essere definito come un apparato complesso, capace di controllare con precisione il moto di un organo meccanico funzionando da interfaccia energetica tra una sorgente di alimentazione elettrica e il carico meccanico finale. Il processo fondamentale che avviene all'interno di tale apparato è la conversione di energia da elettrica a meccanica, che viene modulata in funzione delle specifiche esigenze di coppia e velocità richieste dall'applicazione.

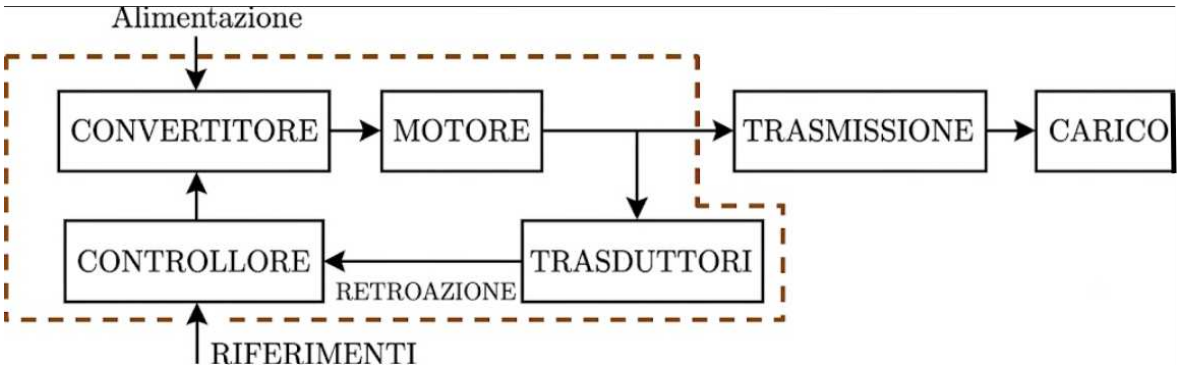


Figura 1.1: Schema a blocchi di un azionamento elettrico

Come illustrato nello schema a blocchi funzionale (Figura 1.1), un moderno azionamento elettrico si compone di quattro sezioni principali, integrate in un anello di controllo chiuso:

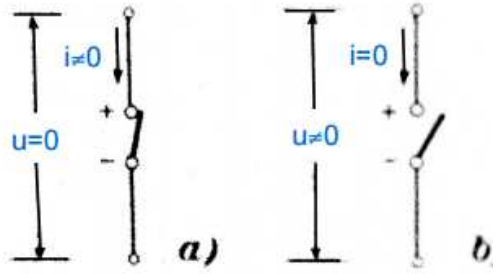
- **Convertitore di Potenza:** Rappresenta l'elemento di attuazione che modula l'energia proveniente dalla sorgente, trasformandola in una forma adatta al motore (ad esempio, modulando tensione e frequenza).
- **Motore Elettrico:** Costituisce il dispositivo di conversione elettromeccanica, in cui l'energia elettrica fornita dal convertitore viene trasformata in lavoro meccanico (coppia e velocità).
- **Trasduttori (Sensori):** Sono elementi essenziali per il sistema di retroazione (feedback). Essi monitorano in tempo reale le grandezze meccaniche (come posizione o velocità) o elettriche (correnti di fase), fornendo le informazioni necessarie al controllo.
- **Controllore:** È il cervello dell'azionamento. Riceve i riferimenti di set-point dall'esterno e le misure provenienti dai trasduttori, elaborando gli algoritmi di controllo necessari per pilotare il convertitore affinché il carico segua il profilo di moto desiderato.

1.1.1 Convertitore di Potenza

Il convertitore di potenza rappresenta l'elemento cardine dell'azionamento, incaricato di modulare l'energia elettrica proveniente dalla rete o da una sorgente in continua, per adattarla alle necessità dinamiche del motore.

La moderna architettura dei convertitori si basa sull'impiego di **convertitori statici**, definiti tali poiché non includono organi meccanici in movimento. A differenza di soluzioni storiche (come i convertitori rotanti tipo Ward-Leonard), i dispositivi statici utilizzano interruttori elettronici allo stato solido — quali diodi, transistori (IGBT, MOSFET) e tiristori — per operare in regime di commutazione (*switch mode*).

- **Efficienza energetica:** Il vantaggio principale di operare in commutazione risiede nel rendimento. Un interruttore ideale, infatti, non è mai sede di perdite di potenza: quando è aperto la corrente è nulla ($i = 0$), mentre quando è chiuso la tensione ai suoi capi è nulla ($u = 0$). In entrambi gli stati, il prodotto $p = u \cdot i$ è nullo, minimizzando la dissipazione energetica rispetto a un funzionamento in zona lineare.
- **Gestione delle armoniche:** La commutazione rapida comporta inevitabilmente la generazione di forme d'onda ricche di componenti armoniche indesiderate. Per garantire un funzionamento regolare del motore e ridurre le interferenze elettromagnetiche, i convertitori integrano spesso induttori e condensatori agendo da filtri.



(a) Switch mode di un interruttore ideale

(b) Componenti dei convertitori statici

Figura 1.2: Convertitori statici

Le topologie di conversione si differenziano in base alla natura della sorgente e del carico:

1. **Chopper (DC-DC):** Per la conversione di tensione continua in tensione continua.
2. **Raddrizzatori (AC-DC):** Per convertire la corrente alternata di rete in continua.
3. **Invertitori (DC-AC):** Fondamentali per gli azionamenti a velocità variabile, permettono di generare tensioni alternate a frequenza e ampiezza variabile a partire da una sorgente continua.

Questa versatilità operativa ha decretato il successo dei convertitori statici, rendendoli la soluzione standard grazie alla loro economicità, alla flessibilità di controllo e all'elevata affidabilità in ambito industriale.

1.1.2 Motore Elettrico

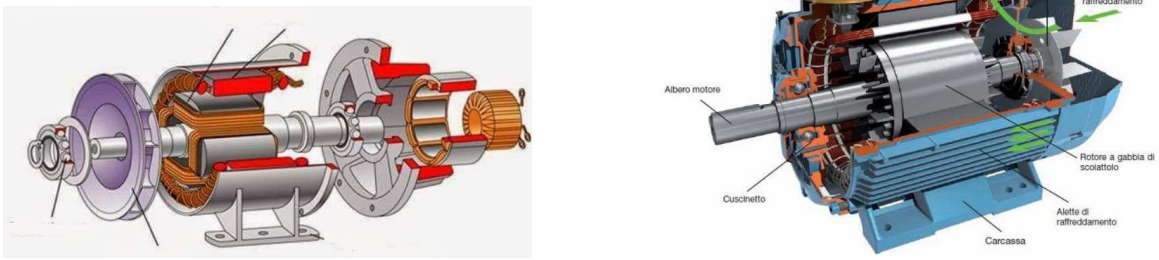
Il motore elettrico costituisce l'elemento attuatore del sistema, il cui compito è la trasformazione dell'energia elettrica modulata dal convertitore in energia meccanica. La scelta della tipologia di macchina è determinante per le prestazioni dinamiche, la complessità del sistema di controllo e l'efficienza complessiva dell'azionamento.

Le macchine elettriche rotanti si suddividono principalmente in due grandi famiglie, distinte in base alla modalità di generazione del campo magnetico:

- **Macchine Asincrone (a induzione):** Caratterizzate da un rotore in cui le correnti sono indotte dal campo magnetico rotante dello statore. Sono ampiamente utilizzate nell'industria pesante per la loro estrema robustezza, semplicità costruttiva e affidabilità, pur richiedendo algoritmi di controllo (come il controllo vettoriale) più complessi per gestire la dipendenza della coppia dallo scorrimento.

- **Macchine Sincrone:** In queste macchine, il rotore ruota in sincronia con il campo magnetico generato dallo statore. Questa famiglia include i **motori a magneti permanenti (PMSM)**, in cui il flusso rotorico è generato da magneti ad alta induzione. Grazie all'elevata densità di coppia, al rendimento superiore e alla possibilità di dimensioni compatte, i motori sincroni a magneti permanenti rappresentano oggi la tecnologia di elezione per applicazioni ad alte prestazioni, come la robotica, i veicoli elettrici e l'automazione di precisione.

Data la rilevanza di tali macchine nei sistemi di controllo moderno, la presente tesi focalizzerà l'attenzione in particolar modo sui motori sincroni a magneti permanenti, analizzandone nel **Capitolo 2** la modellazione matematica, le peculiarità costruttive e le equazioni di stato che ne governano il funzionamento dinamico.



(a) Motore sincrono **(b)** Motore asincrono

Figura 1.3: Motori elettrici

1.1.3 Trasduttori e Sensori

Il sistema di retroazione (feedback) costituisce la base indispensabile per ogni architettura di controllo a ciclo chiuso. La qualità delle prestazioni dinamiche di un azionamento elettrico è intrinsecamente legata all'accuratezza, alla risoluzione e alla velocità di risposta dei segnali acquisiti dai trasduttori. Questi componenti hanno il compito di monitorare in tempo reale le variabili di stato del motore, convertendo grandezze fisiche — meccaniche o elettriche — in segnali elettrici elaborabili dal controllore.

Nell'ambito degli azionamenti ad alte prestazioni, si distinguono due tipologie principali di variabili monitorate:

- **Grandezze Meccaniche:** Per il controllo del moto, è essenziale la conoscenza della posizione angolare del rotore (θ) e della sua velocità (ω). A tale scopo si utilizzano comunemente *encoder* incrementali o assoluti, oppure *resolver*, sensori elettromeccanici particolarmente robusti che permettono di determinare la posizione rotorica con elevata precisione, garantendo affidabilità anche in condizioni ambientali gravose.

- **Grandezze Elettriche:** Il controllo delle correnti di fase (fondamentale per tecniche come il controllo vettoriale) richiede l'acquisizione delle correnti di statore. Tale operazione viene eseguita mediante sensori di corrente (basati ad esempio sull'effetto Hall o su resistori di shunt), che misurano le correnti istantanee fornendo un feedback immediato al controllore.

L'acquisizione di tali dati deve avvenire con una latenza estremamente ridotta. L'integrazione tra i segnali analogici provenienti dai sensori e la logica digitale del controllore avviene solitamente tramite convertitori analogico-digitali (ADC) ad alta velocità. La precisione di tale conversione e la frequenza di campionamento del segnale sono fattori critici: un ritardo nella lettura o un'eccessiva rumorosità del segnale acquisito possono infatti compromettere la stabilità del sistema, limitando la banda passante del controllore e, di conseguenza, la dinamica complessiva dell'azionamento.

La scelta della sensoristica, unitamente alle strategie di filtraggio digitale implementate per ripulire i segnali acquisiti da disturbi elettromagnetici, rappresenta dunque un elemento di design fondamentale per garantire la robustezza dell'intero sistema di controllo.

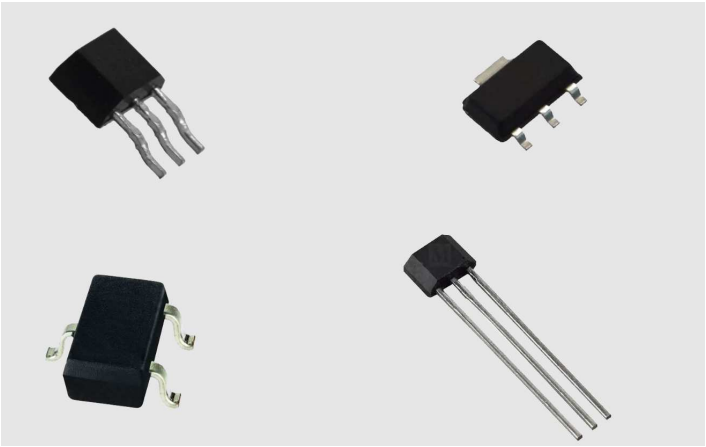


Figura 1.4: Sensori di corrente ad effetto Hall montati sull'inverter

1.1.4 Sistema di Controllo

Il controllo di un moderno azionamento elettrico richiede un'architettura in grado di gestire in tempo reale le complesse dinamiche elettromeccaniche del motore. Il controllore agisce come unità decisionale, elaborando le informazioni provenienti dai sensori per imporre le tensioni ottimali al convertitore di potenza. Ad un efficace sistema di controllo si chiede di soddisfare numerosi requisiti prestazionali, affrontando sfide tecniche che spaziano dalla dinamica del sistema alla sua implementazione hardware:

- **Inseguimento del riferimento:** Il controllore deve garantire che le grandezze di uscita del sistema — principalmente coppia e velocità — seguano il profilo di riferimen-

to imposto con la massima precisione, minimizzando l'errore a regime stazionario e riducendo il tempo di assestamento.

- **Risposta dinamica e reiezione dei disturbi:** Un azionamento moderno deve presentare un'elevata prontezza nel rispondere a variazioni repentine del set-point, mantenendo al contempo la stabilità del sistema in presenza di perturbazioni esterne o variazioni del carico applicato.

- **Gestione operativa e dei vincoli:** Il sistema deve essere in grado di operare entro i margini di sicurezza imposti dai componenti fisici (come i limiti di corrente degli inverter e le tensioni di bus), evitando saturazioni indesiderate e garantendo l'integrità dei componenti elettromeccanici.

- **Implementazione su Hardware Dedicato:** L'esecuzione di tali compiti in tempo reale richiede microcontrollori ad alte prestazioni (quali la famiglia Texas Instruments impiegata in questo lavoro), capaci di coniugare elevate capacità di calcolo in virgola mobile (FPU) con una gestione deterministica degli interrupt, essenziale per mantenere costante la frequenza di campionamento e di modulazione PWM.

La progettazione di tale architettura richiede un attento bilanciamento tra l'accuratezza del modello matematico del sistema e le risorse computazionali disponibili. L'approccio scelto per risolvere tali problematiche, unitamente all'algoritmo di controllo specifico implementato, sarà oggetto di analisi approfondita nel **Capitolo 3**. In tale sede verranno esposti la formulazione del problema di controllo, le strategie adottate per la gestione dei vincoli e le soluzioni implementative finalizzate alla validazione sperimentale del sistema.

Figura 1.5: Microcontrollore Texas Instruments

1.2 Obiettivi della tesi

L'obiettivo principale del presente lavoro di tesi è l'analisi, lo sviluppo e l'implementazione di una strategia di controllo avanzata applicata a un azionamento elettrico. In un contesto industriale sempre più orientato verso la massimizzazione dell'efficienza energetica e la riduzione dei tempi di risposta dinamica, i metodi di controllo convenzionali risultano talvolta limitati nel gestire la natura non lineare e i vincoli operativi tipici dei moderni motori sincroni a magneti permanenti. Per questo, la strategia adottata si concentra su un algoritmo di controllo basato su modello, integrando nativamente i vincoli operativi direttamente nella legge di controllo, garantendo prestazioni ottimali senza superare i limiti strutturali dell'azionamento.

Il lavoro descrive il processo completo di progettazione: dalla modellistica matematica del motore PMSM, allo sviluppo dell'algoritmo in ambiente di simulazione, fino alla verifica sperimentale su piattaforma basata su microcontrollore della famiglia Texas Instruments, dimostrando la fattibilità pratica e l'efficacia del controllo vincolato in contesti applicativi reali.

Le sfide principali che hanno motivato questa ricerca risiedono nelle seguenti necessità:

- **Precisione operativa:** Ottenere un inseguimento del riferimento di coppia e velocità estremamente accurato, minimizzando l'errore a regime stazionario anche in presenza di variazioni repentine del carico.
- **Robustezza ai vincoli:** Garantire il rispetto dei limiti fisici del sistema (correnti massime, tensioni di bus, variazioni dello sforzo di controllo) attraverso una gestione dinamica delle variabili di stato.
- **Efficienza computazionale:** Implementare algoritmi di controllo che, pur essendo complessi, siano compatibili con le risorse di calcolo limitate dei microcontrollori dedicati al controllo motore in tempo reale.

L'attività svolta durante il periodo di tirocinio ha permesso di affrontare tali criticità mediante l'adozione di tecniche di controllo predittivo, focalizzandosi sull'ottimizzazione del comportamento del sistema su un orizzonte temporale definito, trasformando così il problema del controllo in un problema di ottimizzazione vincolata.

1.2.1 Struttura della tesi

Il lavoro è organizzato in modo da fornire una guida dall'analisi teorica dei componenti fondamentali fino alla validazione sperimentale dell'algoritmo di controllo sviluppato. La struttura dei capitoli è la seguente:

- **Capitolo 1:** Introduzione generale sul ruolo degli azionamenti elettrici nell'automazione moderna e definizione degli obiettivi di ricerca.
- **Capitolo 2:** Analisi dettagliata del motore elettrico brushless (BLDC/PMSM), con un focus sulla modellazione matematica, sulle equazioni di stato del sistema e sulle dinamiche elettromeccaniche che governano il comportamento del rotore.
- **Capitolo 3:** Presentazione del sistema di controllo, con una descrizione approfondita dell'architettura dell'MPC e dell'algoritmo di ottimizzazione. Vengono analizzati i criteri di scelta dei pesi nella funzione di costo, la parametrizzazione tramite funzioni di Laguerre e la formulazione matematica del solutore.
- **Capitolo 4:** Implementazione dell'ambiente di sviluppo: descrizione dettagliata della fase di calcolo *offline* tramite MATLAB, della generazione delle *Look-up Tables* per il *Gain Scheduling* e della struttura del blocco Simulink per l'esecuzione in tempo reale, inclusa l'integrazione dello stimatore *Sliding Mode* per la resistenza di statore.
- **Capitolo 5:** Validazione sperimentale e analisi dei risultati: descrizione del setup di laboratorio, della piattaforma hardware basata su architetture Texas Instruments e discussione critica dei risultati ottenuti, con un confronto diretto tra le risposte dinamiche in simulazione e quelle acquisite sul banco prova reale.
- **Capitolo 6:** Conclusioni finali: sintesi delle prestazioni raggiunte, valutazione dell'efficacia dell'approccio proposto e analisi delle possibili direzioni future per l'ulteriore ottimizzazione del sistema di controllo studiato.

Capitolo 2

Modellazione del Motore Sincrono a Magneti Permanenti (PMSM)

Nell'industria dell'automazione moderna, la richiesta di sistemi caratterizzati da elevata efficienza, affidabilità e dinamica operativa ha reso i motori sincroni a magneti permanenti (PMSM), comunemente noti come motori brushless, la tecnologia di riferimento. Rispetto alle tradizionali macchine a corrente continua, questi motori eliminano i contatti striscianti tra statore e rotore, riducendo drasticamente l'usura meccanica e la necessità di manutenzione periodica. Il motore brushless si classifica in due categorie:

- **DC brushless:** caratterizzato da un andamento trapezoidale delle forze elettromotrici indotte che si producono tra due morsetti di statore durante la rotazione.
- **AC brushless:** caratterizzato da un andamento sinusoidale delle forze controelettromotrici indotte.

Il funzionamento del motore brushless si basa sulla commutazione elettronica delle fasi statoriche, gestita tramite inverter in funzione della posizione angolare del rotore — rilevata tramite sensori a effetto Hall o algoritmi sensorless. Questa architettura consente di eliminare attriti e scintillii, garantendo prestazioni costanti anche in ambienti gravosi. La struttura tipica prevede uno statore con avvolgimenti trifase, che genera un campo magnetico rotante, e un rotore a magneti permanenti, il cui numero di poli e disposizione determinano le prestazioni dinamiche del sistema, quali coppia e velocità. L'interazione tra il campo statorico e il flusso rotorico, orchestrata dalla commutazione elettronica, permette una rotazione fluida e un controllo preciso dell'erogazione della coppia motrice. Il campo di applicazione degli azionamenti brushless è circoscritto alle piccole e medie potenze (tipicamente inferiori a 50 kW). Tale limitazione è dettata dalle dimensioni dei magneti permanenti necessari per valori di potenza superiori; inoltre, i costi di realizzazione risultano elevati a causa dell'impiego di materiali magnetici a base di terre rare.

Il presente capitolo è dedicato alla modellazione matematica di tali macchine, con l'obiettivo di fornire le basi teoriche necessarie allo sviluppo di strategie di controllo avanzate. La trattazione si articola su due pilastri fondamentali:

- **Definire le dinamiche elettromeccaniche:** Derivare le equazioni differenziali che descrivono il comportamento del sistema, ponendo particolare attenzione sul legame tra correnti di statore, flusso dei magneti permanenti e coppia sviluppata.
- **Semplificare il modello per il controllo:** Introdurre la trasformata di Park per passare da un sistema di riferimento trifase, variante nel tempo, a un sistema bilineare equivalente sugli assi dq solidali con il rotore. Questo passaggio è condizione necessaria per consentire strategie di controllo vettoriale avanzate, capaci di regolare la macchina con la precisione di un motore a corrente continua a eccitazione indipendente.

La trattazione si baserà su un modello di macchina sincrona trifase, assumendo ipotesi semplificative standard quali la linearità del circuito magnetico (trascurando saturazione e isteresi) e la simmetria degli avvolgimenti statorici. Tali assunzioni, pur semplificando la complessità intrinseca delle macchine elettriche, permettono di ottenere un modello accurato e adatto alla sintesi dei regolatori di corrente che verranno discussi nel prosieguo del lavoro.

2.1 Trasformata di Park

L'analisi dei sistemi elettrici trifase, come gli avvolgimenti statorici di un motore brushless, presenta una complessità intrinseca dovuta all'accoppiamento magnetico tra le fasi e alla variabilità nel tempo delle induttanze in base alla posizione del rotore. In regime dinamico, le equazioni che descrivono tali interazioni risultano analiticamente complesse da risolvere. Per superare questo ostacolo, si ricorre alla **Trasformata di Park**, una tecnica matematica che permette di proiettare le variabili fisiche trifase $(1,2,3)$ su un sistema di riferimento ortogonale, solitamente indicato con gli assi dq (diretto e in quadratura).

2.1.1 Modello Bifase Equivalente

Il primo passo consiste nella semplificazione dell'avvolgimento trifase in un sistema bifase equivalente. Sotto opportune ipotesi di linearità del circuito magnetico (trascurando saturazione e isteresi) e simmetria geometrica, è possibile dimostrare che un sistema di equazioni trifase è algebricamente equivalente a un sistema composto da due avvolgimenti mutuamente disaccoppiati, più un avvolgimento monofase di sequenza zero legato alla componente omopolare.

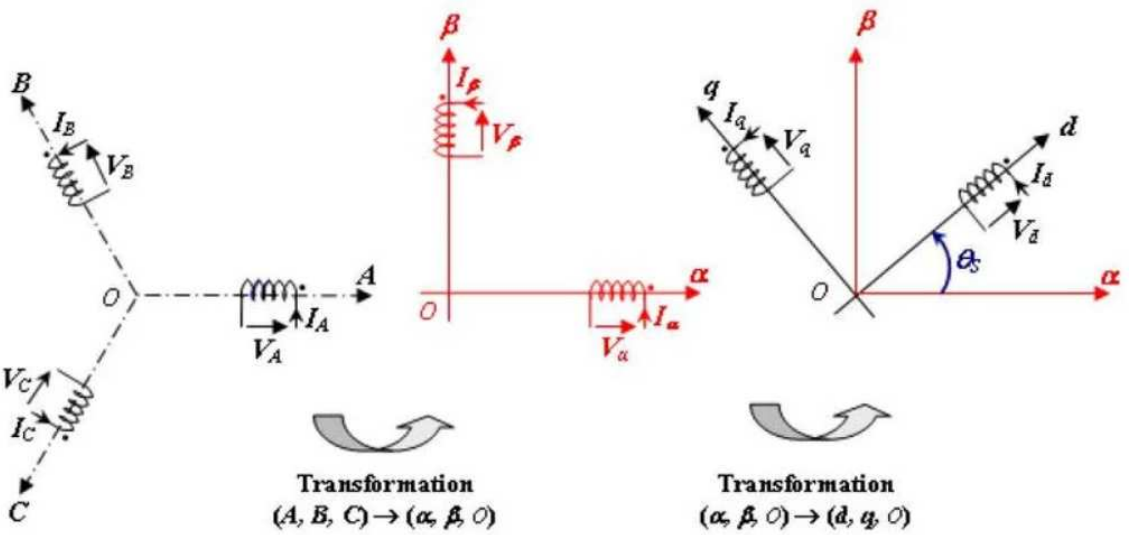


Figura 2.1: Proiezioni assi trifase su un sistema di riferimento ortogonale

2.1.2 Passaggio al sistema di riferimento ruotante

La trasformata di Park permette di passare da un sistema di riferimento fisso (assi $\alpha\beta$) a un sistema di riferimento ruotante (assi dq), solidale con il rotore. Tale trasformazione si ottiene operando una rotazione degli assi di un angolo θ pari alla posizione elettrica del rotore rispetto all'asse di fase 1:

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = T(\theta) \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{2.1}$$

dove $T(\theta)$ è la matrice di trasformazione definita come:

$$T(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \cos(\theta - \frac{2}{3}\pi) & \cos(\theta + \frac{2}{3}\pi) \\ -\sin\theta & -\sin(\theta - \frac{2}{3}\pi) & -\sin(\theta + \frac{2}{3}\pi) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.2}$$

2.1.3 Vantaggi della trasformazione

L'impiego della trasformata di Park introduce benefici fondamentali per il controllo dell'azionamento:

- **Disaccoppiamento:** In un sistema dq correttamente orientato, le correnti di asse diretto (i_d) e in quadratura (i_q) agiscono come variabili indipendenti, analogamente a quanto avviene in una macchina a corrente continua a eccitazione indipendente.
- **Stazionarietà:** Se il sistema di riferimento ruota alla stessa velocità del campo magnetico generato dal rotore, le variabili v_d, v_q, i_d, i_q risultano costanti durante il funziona-

mento a regime. Ciò elimina la dipendenza dal tempo delle equazioni, semplificando enormemente la sintesi dei regolatori.

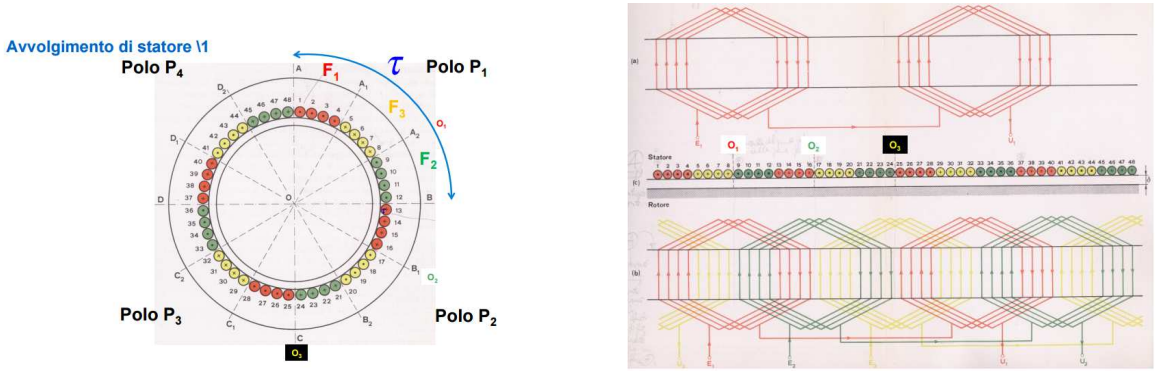
- **Efficienza:** La trasformazione è ortonormale, il che garantisce la conservazione della potenza elettrica tra il sistema trifase e quello trasformato, permettendo un'analisi energetica coerente.

Questa base matematica costituisce il prerequisito essenziale per il **Capitolo 2.2**, dove le equazioni del motore saranno espresse nel sistema dq per permettere l'implementazione dell'algoritmo di controllo.

2.2 Modellazione del Motore Sincrono a Magneti Permanenti (PMSM)

Il motore sincrono a magneti permanenti (PMSM), o *brushless*, è una macchina elettrica caratterizzata da una struttura in cui lo statore, analogo a quello di un motore asincrono, ospita un avvolgimento trifase simmetrico ed equilibrato, mentre il rotore presenta magneti permanenti che fungono da induttore.

Lo statore è composto da un pacco lamellare ferromagnetico con avvolgimenti distribuiti (o concentrati) lungo la periferia interna, progettati per generare un campo magnetico rotante di forma sinusoidale.



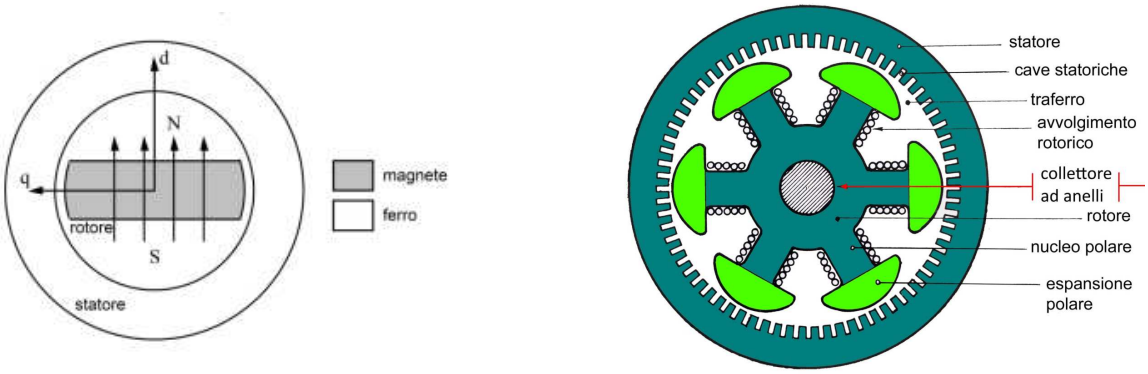
(a) Cave dello statore **(b)** Avvolgimenti dello statore

Figura 2.2: Statore motore brushless

Per quanto concerne il rotore, si distinguono due tipologie strutturali principali in base alla disposizione dei magneti e alla natura del traferro:

- **Struttura isotropa:** Il rotore presenta un traferro costante, con induttanze di asse diretto e quadratura uguali ($L_d = L_q$). Tale configurazione garantisce un comportamento lineare e una coppia dipendente esclusivamente dalla corrente in quadratura.

- **Struttura anisotropa (a poli salienti):** Il traferro non è costante, conferendo alla macchina una riluttanza variabile in funzione della posizione angolare del rotore. In questo caso, le induttanze dipendono dalla posizione rotorica θ_m, introducendo un termine aggiuntivo di *coppia di riluttanza* dovuto alla differenza tra le induttanze di asse L_d e L_q.



(a) Rotore anisotropo **(b)** Rotore isotropo

Figura 2.3: Rotore motore brushless

2.2.1 Relazioni elettromagnetiche e transizione al sistema rotante

Il comportamento elettromagnetico della macchina nel dominio delle fasi $(1,2,3)$ è descritto da un sistema di equazioni differenziali lineari, espresse in forma matriciale come segue:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix} \tag{2.3}$$

Il vettore dei flussi concatenati $\underline{\varphi}_s$ dipende sia dalle correnti statoriche, attraverso la matrice delle induttanze $L_s(\theta_m)$, sia dal flusso generato dai magneti permanenti rotanti $\underline{\varphi}_{mp}(\theta_m)$:

$$\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix} = L_s(\theta_m) \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} + \begin{bmatrix} \varphi_{mp,1}(\theta_m) \\ \varphi_{mp,2}(\theta_m) \\ \varphi_{mp,3}(\theta_m) \end{bmatrix} \tag{2.4}$$

La matrice $L_s(\theta_m)$ è simmetrica e le sue componenti dipendono esplicitamente dalla posizione del rotore θ_m. In presenza di un rotore anisotropo, il circuito magnetico vede una riluttanza variabile, rendendo tale dipendenza ancora più marcata. Infine, la dinamica meccanica è governata dall'equazione del moto:

$$\frac{d\dot{\theta}_m}{dt} = \frac{p}{J}(C_m - C_r) \tag{2.5}$$

dove p è il numero di coppie polari, J l'inerzia, C_m la coppia motrice e C_r la coppia resistente.

Il modello espresso nelle equazioni precedenti, presenta una criticità operativa: le mutue-induttanze e le auto-induttanze dipendono dalla posizione del rotore θ_m, rendendo il sistema di equazioni differenziali a coefficienti variabili nel tempo. Questa dipendenza non è solo un ostacolo analitico, ma nasconde l'essenza della dinamica del motore.

Per superare tale limite, si applica la **Trasformata di Park**. L'idea fisica sottostante è quella di cambiare il punto di osservazione: anziché osservare i campi magnetici da un riferimento fisso allo statore, ci posizioniamo sul rotore. In questo sistema di riferimento, solidale con il flusso del magnete permanente (asse d), la macchina appare elettricamente stazionaria.

Proiettando le grandezze trifasi nel riferimento rotante dq, le equazioni di tensione si trasformano in un sistema accoppiato che rivela chiaramente i fenomeni fisici in gioco:

$$\begin{cases} v_{sd} = R_s i_{sd} + \frac{d\varphi_{sd}}{dt} - \omega_m \varphi_{sq} \\ v_{sq} = R_s i_{sq} + \frac{d\varphi_{sq}}{dt} + \omega_m \varphi_{sd} \end{cases} \tag{2.6}$$

con i flussi concatenati definiti da:

$$\begin{cases} \varphi_{sd} = L_d i_{sd} + \varphi_{mp} \\ \varphi_{sq} = L_q i_{sq} \end{cases} \tag{2.7}$$

I termini $\pm\omega_m\varphi$ rappresentano le forze contro-elettromotrici (f.c.e.m.) di rotazione. Essi introducono un accoppiamento incrociato tra gli assi d e q: una variazione di corrente su un asse influenza la tensione sull'altro. Questo fenomeno è l'analogo matematico della rotazione di un campo magnetico in una macchina a corrente continua.

Per semplificare ulteriormente l'analisi dei transitori, è estremamente utile ricorrere alla notazione complessa. Definendo il vettore spaziale di corrente $\bar{i}_s = i_{sd} + j i_{sq}$ e di flusso $\bar{\varphi}_s = \varphi_{sd} + j\varphi_{sq}$, il sistema elettrico si condensa in una singola equazione vettoriale:

$$\bar{v}_s = R_s \bar{i}_s + \frac{d\bar{\varphi}_s}{dt} + j\omega_m \bar{\varphi}_s \tag{2.8}$$

Questa forma compatta permette di interpretare il motore come un circuito nel piano complesso, dove la velocità angolare ω_m agisce come un operatore di rotazione. Tale visualizzazione sarà la base per comprendere come, agendo opportunamente sulle correnti i_{sd} e i_{sq}, sia possibile regolare la coppia con la stessa dinamica di un motore a spazzole, ma con la robustezza di una macchina sincrona.

2.2.2 Equazione della coppia e analogia con la macchina a collettore

La coppia elettromagnetica C_m generata dal motore brushless può essere ricavata analizzando il bilancio delle potenze attive nel dominio trasformato di Park. Considerando la potenza elettromagnetica $P_m(t)$ come la potenza convertita dal sistema elettrico a quello meccanico, si ottiene:

$$P_m(t) = \frac{3}{2}\left[\omega_m(\varphi_{sd} i_{sq} - \varphi_{sq} i_{sd})\right] \tag{2.9}$$

Sostituendo le espressioni dei flussi $\varphi_{sd} = L_d i_{sd} + \varphi_{mp}$ e $\varphi_{sq} = L_q i_{sq}$, l'espressione della coppia $C_m = P_m/\omega_m$ risulta:

$$C_m = \underbrace{\frac{3}{2} p \varphi_{mp} i_{sq}}_{\text{Coppia di magnete}} + \underbrace{\frac{3}{2} p (L_d - L_q) i_{sd} i_{sq}}_{\text{Coppia di riluttanza}} \tag{2.10}$$

Questa equazione giustifica la strategia del **controllo vettoriale**:

- Se la macchina è **isotropa** ($L_d = L_q$), la coppia di riluttanza si annulla. In questo caso, il motore si comporta esattamente come una macchina a corrente continua a eccitazione indipendente: la corrente i_{sd} controlla il flusso (che è fisso, dato dai magneti), mentre la corrente i_{sq} controlla linearmente la coppia.
- Se la macchina è **anisotropa** ($L_d \neq L_q$), la coppia di riluttanza fornisce un contributo aggiuntivo che può essere sfruttato per incrementare le prestazioni, complicando tuttavia la legge di controllo.

2.3 Azionamenti con motore AC brushless

Gli azionamenti basati su motori AC brushless rappresentano la soluzione industriale di riferimento per applicazioni che richiedono elevate prestazioni dinamiche. In tali sistemi, la distribuzione uniforme degli avvolgimenti statorici, sfasati di 120°, garantisce una concatenazione del flusso φ_{mp} di tipo sinusoidale, permettendo l'adozione di sofisticate tecniche di controllo vettoriale.

Il controllo vettoriale si basa sull'orientamento degli assi dq in modo che siano solidali con il rotore. Nello specifico, si impone l'allineamento dell'asse d con il flusso prodotto dai magneti permanenti (φ_{mp}).

Una scelta fondamentale nella strategia di controllo è l'imposizione di una componente di corrente diretta nulla:

$$i_{sd} = 0 \tag{2.11}$$

Tale condizione è ottimale per i motori isotropi, poiché permette di minimizzare il modulo del vettore corrente $\bar{i}_s$ a parità di coppia generata, evitando inutili perdite per effetto Joule sull'asse di flusso. Sotto tale ipotesi e considerando il sistema in regime stazionario ($\frac{d}{dt} = 0$), le equazioni elettriche e i flussi concatenati si semplificano nel seguente modo:

$$\begin{cases} v_{sd} = -\omega_m \varphi_{sq} \\ v_{sq} = R_s i_{sq} + \frac{d\varphi_{sq}}{dt} + \omega_m \varphi_{sd} \end{cases} \tag{2.12}$$

$$\begin{cases} \varphi_{sd} = \varphi_{mp} \\ \varphi_{sq} = L_q i_{sq} \end{cases} \tag{2.13}$$

$$C_m = \frac{3}{2} p \varphi_{mp} i_{sq} \tag{2.14}$$

In questo modo, il comportamento del motore sincrono diviene analogo a quello di una macchina a corrente continua, dove la corrente in quadratura i_{sq} assume il ruolo di corrente di armatura ma non è possibile agire sull'eccitazione (deflussaggio) poiché $i_{sd} = 0$

2.4 Campo di operatività

La strategia di controllo deve adattarsi al variare della velocità meccanica $\dot{\theta}_m$. Si distinguono tre regioni operative principali:

1. **Regione a coppia costante** ($0 \leq \dot{\theta}_m < \omega_b$) Per velocità inferiori alla velocità base ω_b, si mantiene $i_{sd} = 0$ e si eroga la massima corrente consentita $i_{sq} = i_{max}$. In questo intervallo, il motore lavora a coppia costante e pari al suo valore massimo:

$$C_{m,max} = p \varphi_{mp} i_{max} \tag{2.15}$$

2. **Regione di indebolimento di campo** ($\omega_b \leq \dot{\theta}_m < \omega^*$) Al superare della velocità base, la forza contro-elettromotrice $E = \omega_m \varphi_{mp}$ tende a eguagliare la tensione massima erogabile dall'inverter ($V_{s,max}$). Per aumentare ulteriormente la velocità, è necessario introdurre una componente di corrente diretta negativa ($i_{sd} < 0$). Questa corrente crea un flusso in opposizione a quello del magnete permanente, permettendo di compensare l'aumento di E e di mantenere V_s entro i limiti. Questa pratica, sebbene impropriamente definita *deflussaggio*, comporta una riduzione di i_{sq} per rispettare il vincolo sulla corrente totale $i_s \leq i_{max}$, causando una diminuzione della coppia disponibile.

3. **Regione a potenza costante** ($\omega > \omega^*$) Oltre la velocità ω^*, la corrente i_{sd} viene aumentata fino al valore limite $-i_{max}$ (velocità a vuoto). In questo regime, trascurando la

caduta resistiva R_s, il sistema evolve seguendo un andamento a potenza costante:

$$C_m \propto \frac{1}{\omega} \tag{2.16}$$

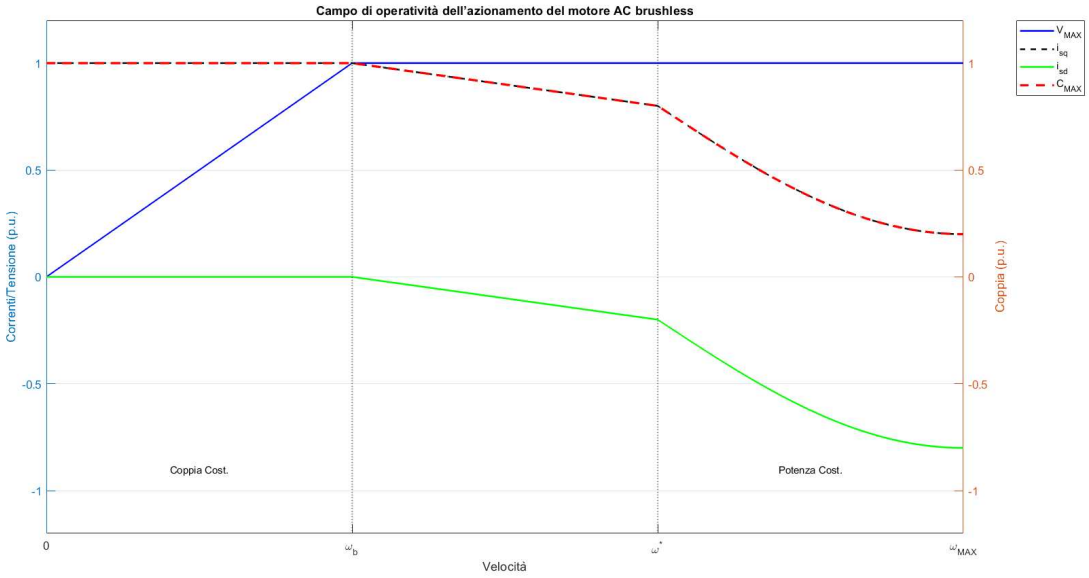


Figura 2.4: Campo di operatività dell'azionamento del motore AC brushless

2.5 Necessità di una strategia di controllo avanzata

La disamina condotta nelle sezioni precedenti ha evidenziato come il comportamento di un motore AC brushless sia intrinsecamente non lineare e caratterizzato da forti accoppiamenti tra le variabili di stato. Sebbene la Trasformata di Park permetta di ricondurre il sistema a una forma più gestibile, rimane la sfida di regolare le correnti di asse i_{sd} e i_{sq} in modo che la macchina segua le dinamiche desiderate, rispettando rigorosamente i vincoli fisici dell'azionamento. Il limite intrinseco delle strategie di controllo tradizionali risiede nella loro difficoltà a gestire contemporaneamente:

- **La natura non lineare:** Le f.c.e.m. di rotazione introducono termini che variano con la velocità, rendendo complesso il mantenimento di prestazioni ottimali in un ampio spettro operativo.
- **La gestione dei vincoli:** In condizioni di transitorio o di funzionamento in deflussaggio, è cruciale non superare i limiti di corrente dell'inverter o i limiti di tensione del bus DC, pena la perdita di sincronismo o il danneggiamento dell'elettronica di potenza.

- **La dinamica del carico:** In molte applicazioni, come il controllo di mandrini o assi di precisione, il carico può presentare variazioni repentine che richiedono un'ottima capacità di risposta.

È proprio per superare tali limitazioni che emerge la necessità di una strategia di controllo di livello superiore. Mentre i metodi tradizionali si limitano a imporre correnti di riferimento senza una visione globale del sistema, l'approccio proposto in questa tesi — basato su un *controllo predittivo vincolato* (Model Predictive Control) — permette di riformulare il problema di controllo come un problema di ottimizzazione.

Attraverso l'impiego del modello matematico derivato in questo capitolo, il controllore è in grado di predire l'evoluzione futura delle variabili di stato e di calcolare l'azione di controllo ottimale che minimizza una funzione di costo definita, garantendo il pieno rispetto dei vincoli fisici del sistema. Il Capitolo 3 sarà dunque dedicato alla formulazione formale di tale algoritmo di controllo, dettagliandone l'architettura e le proprietà di convergenza.

Capitolo 3

Algoritmo di Controllo (Model Predictive Control)

Nonostante il continuo progresso nella teoria del controllo, il controllore PID rimane lo standard industriale per la sua intrinseca robustezza e semplicità di implementazione. Tuttavia, l'architettura di controllo basata su classici loop indipendenti presenta dei limiti strutturali quando applicata a sistemi complessi e fortemente accoppiati, come gli azionamenti elettrici ad alte prestazioni. Tali limitazioni si manifestano principalmente in due aree: la mancanza di un coordinatore gerarchico in grado di gestire in modo sistematico gli obiettivi di performance globale e, soprattutto, l'incapacità di integrare nativamente i vincoli fisici e operativi del processo — quali le correnti di saturazione dell'inverter o i limiti di tensione del bus DC — direttamente all'interno della legge di controllo.

In questo contesto, il controllo predittivo basato su modello (Model Predictive Control, MPC) emerge come la soluzione ideale per superare tali criticità. Grazie alla sua formulazione matematica, l'MPC non si limita a reagire all'errore, ma ottimizza il comportamento del sistema su un orizzonte temporale definito, gestendo in modo naturale e sistematico un'ampia varietà di vincoli operativi. Proprio per questa capacità di coniugare prestazioni dinamiche superiori e rigoroso rispetto dei limiti di sistema, l'MPC si è affermato come una delle tecniche di controllo avanzato più promettenti per le moderne applicazioni di azionamento industriale. [1]

Il presente capitolo sviluppa la strategia di controllo seguendo un approccio metodologico: partendo dai fondamenti del controllo a orizzonte mobile, verranno analizzate le formulazioni per il controllo vincolato e l'implementazione basata su funzioni ortonormali (come le funzioni di Laguerre). Il percorso di progettazione integra sistematicamente la modellazione del sistema con le tecniche di ottimizzazione, affrontando le sfide legate alla gestione dei vincoli operativi e alla messa a punto dei parametri di controllo. Tali strategie, validate in ambiente di simulazione MATLAB/Simulink (Capitolo 4), forniscono il framework teorico necessario per l'implementazione in tempo reale sull'hardware dedicato (Capitolo 5).

3.1 Modellazione discreta del sistema

La filosofia di progettazione del controllo predittivo basato su modello (MPC) si fonda sulla disponibilità di una rappresentazione matematica accurata dell'impianto. Per il sistema in esame, si è adottato un modello nello spazio degli stati, in quanto tale formulazione garantisce che le informazioni correnti necessarie per prevedere l'evoluzione futura della variabile controllata siano interamente contenute nel vettore di stato al tempo attuale $x(k)$ [1].

La dinamica del sistema viene descritta dall'interazione tra lo stato interno e le azioni di controllo, permettendo al controllore di anticipare il comportamento dell'impianto prima ancora che l'errore si manifesti effettivamente.

Il sistema dinamico in spazio di stato è descritto dalle seguenti equazioni alle differenze:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{3.1}$$

$$y(k) = C_d x(k) \tag{3.2}$$

dove $x(k)$ rappresenta lo stato del sistema all'istante k, $u(k)$ l'ingresso di controllo e $y(k)$ l'uscita misurata.

Poiché l'algoritmo di ottimizzazione MPC viene eseguito dal microcontrollore a intervalli di campionamento regolari (T_s), è necessario discretizzare il modello a tempo continuo derivato nel Capitolo 2. Sebbene la discretizzazione esatta (ZOH) offra una rappresentazione precisa dell'evoluzione dei flussi, per applicazioni di controllo in tempo reale su microcontrollori a risorse limitate, l'approssimazione di Eulero in avanti rappresenta una soluzione efficace e computazionalmente efficiente.

Assumendo un tempo di campionamento T_s sufficientemente piccolo rispetto alla costante di tempo elettrica del sistema ($\tau_e = L/R_s$), la derivata temporale dello stato $\dot{x}(t)$ può essere approssimata come:

$$\dot{x}(t) \approx \frac{x(k+1) - x(k)}{T_s} \tag{3.3}$$

Sostituendo tale approssimazione nelle equazioni differenziali del motore ($\dot{x} = Ax + Bu$), si ottiene la forma discreta:

$$x(k+1) = \underbrace{(I + T_s A)}_{A_d} x(k) + \underbrace{(T_s B)}_{B_d} u(k) \tag{3.4}$$

dove I è la matrice identità di ordine 2. In questa formulazione, il vettore di stato $x(k) = [i_{sd}(k) \quad i_{sq}(k)]^T$ rappresenta le correnti di asse, mentre l'ingresso $u(k) = [v_{sd}(k) \quad v_{sq}(k)]^T$ è costituito dalle tensioni fornite dall'inverter.

Considerando il modello del motore PMSM nel riferimento dq, le matrici del sistema

continuo sono:

$$A = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_m \frac{L_q}{L_d} \\ -\omega_m \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \tag{3.5}$$

Applicando la discretizzazione di Eulero ($A_d = I + T_s A$ e $B_d = T_s B$), otteniamo le matrici discrete:

$$A_d = \begin{bmatrix} 1 - \frac{T_s R_s}{L_d} & \omega_m T_s \frac{L_q}{L_d} \\ -\omega_m T_s \frac{L_d}{L_q} & 1 - \frac{T_s R_s}{L_q} \end{bmatrix}, \quad B_d = \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \end{bmatrix} \tag{3.6}$$

Questa struttura diagonale è particolarmente vantaggiosa: permette di gestire il controllo degli assi d e q come due sistemi scalari indipendenti, semplificando notevolmente il carico computazionale dell'algoritmo di ottimizzazione che verrà implementato sul microcontrollore.

3.1.1 Formulazione del modello a stato esteso

Per conferire al controllore la capacità di eliminare l'errore a regime in presenza di disturbi o incertezze di modellazione, il sistema dinamico viene arricchito definendo un nuovo vettore di stato esteso. Tale approccio si basa sull'integrazione di due componenti chiave: la variazione dello stato e l'errore di inseguimento $e(k) = r(k) - y(k)$.

Definiamo lo stato esteso $\tilde{x}(k)$ come:

$$\tilde{x}(k) = \begin{bmatrix} \Delta x(k) \\ e(k) \end{bmatrix} = \begin{bmatrix} x(k) - x(k-1) \\ r(k) - y(k) \end{bmatrix} \tag{3.7}$$

Dove $\Delta x(k)$ rappresenta la variazione del vettore di stato e $y(k)$ l'uscita attuale. Utilizzando la variabile di controllo incrementale $\Delta u(k) = u(k) - u(k-1)$, le nuove equazioni di stato esteso del sistema diventano:

$$\Delta x(k+1) = A_d \Delta x(k) + B_d \Delta u(k) \tag{3.8}$$

$$y(k+1) = y(k) + C_d \Delta x(k+1) = y(k) + C_d A_d \Delta x(k) + C_d B_d \Delta u(k) \tag{3.9}$$

Combinando (3.8) e (3.9) in una forma matriciale compatta, otteniamo il modello esteso:

$$\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & 0^T \\ C_d A_d & I \end{bmatrix}}_{\tilde{A}} \begin{bmatrix} \Delta x(k) \\ e(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}}_{\tilde{B}} \Delta u(k) \tag{3.10}$$

$$y(k) = \underbrace{\begin{bmatrix} 0 & I \end{bmatrix}}_{\tilde{C}} \begin{bmatrix} \Delta x(k) \\ e(k) \end{bmatrix} \tag{3.11}$$

Motivazioni della scelta

L'impiego del modello esteso e della variabile incrementale $\Delta u(k)$ è giustificato da tre ragioni principali:

- **Azione Integrale:** L'inclusione della variazione di controllo $\Delta u(k)$ all'interno dell'ottimizzazione forza il controllore ad agire come un integratore, annullando naturalmente l'errore di inseguimento a regime stazionario anche in presenza di disturbi di carico costanti.
- **Gestione dei vincoli:** Lavorare con $\Delta u(k)$ permette di imporre limiti fisici non solo sulla tensione massima dell'inverter, ma anche sulla *slew-rate* (velocità di variazione) del segnale di comando, proteggendo i componenti dell'elettronica di potenza.
- **Stabilità e robustezza:** Lo stato esteso permette di includere l'errore di inseguimento $e(k) = r(k) - y(k)$ direttamente nella funzione di costo, facilitando la sintesi di un regolatore che risponda dinamicamente alle variazioni del setpoint $r(k)$.

Questa formulazione trasforma il problema di controllo in un'ottimizzazione dell'incremento di comando, rendendo l'algoritmo intrinsecamente immune ai bias del modello e migliorando significativamente la precisione di inseguimento nel riferimento di corrente per il motore PMSM.

3.2 Fondamenti del Controllo Predittivo (MPC)

L'idea centrale dell'MPC consiste nell'utilizzare il modello matematico del processo per predire il comportamento futuro delle variabili di stato su un prefissato orizzonte temporale, calcolando di conseguenza la sequenza di controllo ottimale tale per cui l'uscita del sistema segua il riferimento desiderato, pur rispettando i vincoli fisici [1]. Per fare ciò, è necessario esprimere la predizione dell'uscita come funzione degli stati correnti e delle future variazioni di controllo.

- **Orizzonte di predizione (N_p):** indica il numero di passi futuri su cui il controllore valuta l'evoluzione dello stato del sistema. Un orizzonte più lungo permette una visione più ampia della dinamica del sistema, migliorando la stabilità ma aumentando l'onere computazionale [2].
- **Orizzonte di controllo (N_c):** rappresenta il numero di campioni in cui l'azione di controllo $u(k)$ viene effettivamente ottimizzata. Solitamente si assume $N_c \leq N_p$, ipotizzando che dopo l'orizzonte di controllo l'ingresso rimanga costante [3].

La sequenza di variazioni di controllo che vogliamo ottimizzare è definita dal vettore $\Delta\mathbf{U}$:

$$\Delta\mathbf{U} = [\Delta u(k|k), \Delta u(k+1|k), \ldots, \Delta u(k+N_c-1|k)]^T \tag{3.12}$$

dove $\Delta u(k+j|k)$ rappresenta la variazione di tensione che intendiamo applicare all'inverter al passo futuro j.

Utilizzando il modello discreto definito nella Sezione 3.1, la traiettoria degli stati futuri può essere predetta ricorsivamente a partire dallo stato corrente $x(k)$:Utilizzando il modello discreto definito nella Sezione 3.1, la traiettoria degli stati futuri può essere predetta ricorsivamente a partire dallo stato corrente $x(k)$. Le predizioni sull'orizzonte di predizione sono date da:

$$\begin{aligned} x(k+1|k) &= A_d x(k) + B_d \Delta u(k|k) \\ x(k+2|k) &= A_d x(k+1|k) + B_d \Delta u(k+1|k) \\ &= A_d^2 x(k) + A_d B_d \Delta u(k|k) + B_d \Delta u(k+1|k) \\ &\vdots \\ x(k+i|k) &= A_d^i x(k) + \sum_{j=0}^{i-1} A_d^{i-j-1} B_d \Delta u(k+j|k) \end{aligned} \tag{3.13}$$

Generalizzando per l'intero orizzonte di predizione N_p, le uscite previste $\mathbf{Y} = [y(k+1|k), y(k+2|k), \ldots, y(k+N_p|k)]^T$ possono essere formulate in forma matriciale compatta come:

$$\mathbf{Y} = \mathbf{F}x(k) + \mathbf{\Phi}\Delta\mathbf{U} \tag{3.14}$$

dove:

- $\mathbf{Y} \in \mathbb{R}^{N_p \times 1}$ è il vettore delle uscite future predette;
- $\Delta\mathbf{U} \in \mathbb{R}^{N_c \times 1}$ è il vettore delle variazioni di controllo ottimizzabili;
- $\mathbf{F} \in \mathbb{R}^{N_p \times n}$ è la matrice di evoluzione libera del sistema;
- $\mathbf{\Phi} \in \mathbb{R}^{N_p \times N_c}$ è la matrice di dinamica del controllo (matrice di convoluzione).

Questa formulazione è di importanza cruciale: essa separa l'effetto dello stato attuale $x(k)$ (noto tramite misura) dall'effetto delle future mosse di controllo $\Delta\mathbf{U}$ (le variabili decisionali). La matrice $\mathbf{F}$ cattura la risposta del sistema in assenza di nuovi ingressi, mentre $\mathbf{\Phi}$ descrive come ogni variazione di controllo influenzerà le uscite future [1].

$$\mathbf{F} = \begin{bmatrix} C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{N_p} \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} C_d B_d & 0 & \cdots & 0 \\ C_d A_d B_d & C_d B_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_d A_d^{N_p-1} B_d & C_d A_d^{N_p-2} B_d & \cdots & C_d A_d^{N_p-N_c} B_d \end{bmatrix} \tag{3.15}$$

Tale struttura permette di trasformare il problema di controllo in un'ottimizzazione quadratica standard, poiché l'uscita $\mathbf{Y}$ risulta una funzione lineare e diretta del vettore di controllo $\Delta\mathbf{U}$ che intendiamo calcolare.

L'ottimizzazione viene risolta ad ogni istante di campionamento k seguendo il principio del *Receding Horizon*: viene calcolata l'intera sequenza di controllo, ma solo il primo valore $\Delta u(k|k)$ viene applicato al sistema. Al passo successivo, l'orizzonte viene traslato in avanti e l'intero processo di ottimizzazione viene risolto nuovamente al passo k + 1 utilizzando la misura reale dello stato x(k + 1). In questo modo, il sistema è in grado di compensare automaticamente eventuali errori di modellazione o disturbi esterni non previsti nella predizione iniziale [1]. Il principio dell'orizzonte mobile può essere interpretato come l'esecuzione di una sequenza di ottimizzazioni su finestre temporali che si spostano progressivamente [4].

Il processo decisionale dell'MPC all'interno di ogni finestra di ottimizzazione si articola attraverso una sequenza logica che trasforma il modello dinamico in una legge di controllo operativa:

1. **Predizione del comportamento futuro:** Sfruttando il modello nello spazio degli stati esteso (Sezione 3.1.1), il controllore calcola la traiettoria evolutiva delle variabili di stato e di uscita per l'intero orizzonte di predizione N_p. Come espresso nella relazione $\mathbf{Y} = \mathbf{F}x(k) + \mathbf{\Phi}\Delta\mathbf{U}$, tale predizione è funzione sia dello stato attuale che della futura sequenza di controllo.

2. **Definizione dell'obiettivo:** Si definisce un criterio di ottimizzazione, noto come funzionale di costo J, che quantifica l'errore di inseguimento tra la traiettoria predetta e il riferimento desiderato, includendo opportuni termini di penalizzazione per lo sforzo di controllo [2].

3. **Ottimizzazione vincolata:** Si risolve il problema di minimizzazione di J rispetto al vettore di variazione di controllo $\Delta\mathbf{U}$, operando nel rispetto dei vincoli fisici di sistema (limiti di corrente e tensione). Questa fase trasforma le specifiche di performance in un valore numerico ottimale.

4. **Applicazione e Receding Horizon:** Coerentemente con il principio del *receding horizon*, solo il primo incremento di controllo $\Delta u(k|k)$ della sequenza ottimale viene effet-

tivamente applicato al sistema. All'istante di campionamento successivo, il processo si ripete integrando le nuove informazioni di stato misurate.

Questa struttura algoritmica, in cui la predizione guida l'ottimizzazione, costituisce il framework teorico che verrà dettagliato nel seguito, a partire dalla formulazione analitica del funzionale di costo J [3].

3.3 Formulazione del funzionale di costo

Per determinare la sequenza di controllo ottimale, l'algoritmo MPC minimizza un funzionale di costo J, tipicamente di tipo quadratico, che esprime il compromesso tra la precisione dell'inseguimento del riferimento e il consumo di energia dell'attuatore. Nel contesto di un sistema di controllo del motore PMSM, la funzione obiettivo è definita come [1]:

$$J = (\mathbf{R_s} - \mathbf{Y})^T \mathbf{Q} (\mathbf{R_s} - \mathbf{Y}) + \Delta \mathbf{U}^T \mathbf{R} \Delta \mathbf{U} \tag{3.16}$$

dove:

- $\mathbf{R_s}$ è il vettore del riferimento desiderato (setpoint delle correnti i_{sd}^* e i_{sq}^*) esteso sull'orizzonte di predizione N_p.
- $\mathbf{Y}$ è il vettore delle uscite predette ottenuto dalla relazione $\mathbf{Y} = \mathbf{F}x(k) + \mathbf{\Phi}\Delta\mathbf{U}$.
- $\mathbf{Q} \in \mathbb{R}^{N_p \times N_p}$ è la matrice di ponderazione dell'errore, che determina quanto severamente l'algoritmo debba penalizzare lo scostamento dal setpoint.
- $\mathbf{R} \in \mathbb{R}^{N_c \times N_c}$ è la matrice di penalizzazione del segnale di controllo, utilizzata per limitare l'eccessiva variazione delle tensioni e prevenire l'instabilità.

Sostituendo la relazione di predizione $\mathbf{Y}$ all'interno del funzionale di costo, J diviene una forma quadratica nella variabile di ottimizzazione $\Delta\mathbf{U}$:

$$J = (\mathbf{R_s} - \mathbf{F}x(k))^T \mathbf{Q} (\mathbf{R_s} - \mathbf{F}x(k)) - 2\Delta\mathbf{U}^T \mathbf{\Phi}^T \mathbf{Q} (\mathbf{R_s} - \mathbf{F}x(k)) + \Delta\mathbf{U}^T (\mathbf{\Phi}^T \mathbf{Q} \mathbf{\Phi} + \mathbf{R}) \Delta\mathbf{U} \tag{3.17}$$

La minimizzazione di tale funzione, ovvero $\frac{\partial J}{\partial \Delta \mathbf{U}} = 0$, conduce alla legge di controllo ottimale. In assenza di vincoli, la soluzione analitica è data da:

$$\Delta\mathbf{U}^* = (\mathbf{\Phi}^T \mathbf{Q} \mathbf{\Phi} + \mathbf{R})^{-1} \mathbf{\Phi}^T \mathbf{Q} (\mathbf{R_s} - \mathbf{F}x(k)) \tag{3.18}$$

Questa formulazione evidenzia come l'MPC operi come un regolatore lineare che, ad ogni passo, calcola la variazione di comando ottimale basandosi sulla distanza tra lo stato predetto "libero" ($\mathbf{F}x(k)$) e il riferimento desiderato ($\mathbf{R_s}$).

3.4 Gestione dei vincoli operativi

L'applicazione del controllo MPC agli azionamenti elettrici richiede il rispetto dei limiti fisici del sistema, quali la saturazione della tensione dell'inverter e la limitazione sulla variazione dello sforzo di controllo. Tali vincoli vengono imposti sulla variabile manipolabile $\Delta\mathbf{U}$ e sul comando assoluto $\mathbf{U}$:

$$\begin{cases} \Delta\mathbf{U}_{min} \leq \Delta\mathbf{U} \leq \Delta\mathbf{U}_{max} \\ \mathbf{U}_{min} \leq \mathbf{U} \leq \mathbf{U}_{max} \end{cases} \tag{3.19}$$

Per il motore PMSM, i vincoli principali sono:

1. **Saturazione di tensione:** Il limite di tensione dell'inverter è definito dal vincolo $|\mathbf{v}_{dq}| \leq V_{dc}/\sqrt{3}$. Tuttavia, al fine di massimizzare la dinamica di coppia, si è adottata una strategia di saturazione asimmetrica (prioritizzazione dell'asse q). Invece di un vincolo circolare rigido, si impone una restrizione che garantisce priorità alla componente di coppia v_{sq} rispetto alla componente di flusso v_{sd}:

$$v_{sq,max} = \varepsilon_q \cdot R, \quad v_{sd,max} = \sqrt{R^2 - v_{sq,max}^2} \tag{3.20}$$

   dove $R = V_{dc}/\sqrt{3}$ rappresenta il raggio del cerchio di tensione e $\varepsilon_q \in (0,1)$ è il coefficiente di prioritizzazione. Questa tecnica permette di mantenere il controllo della coppia anche in condizioni di tensione limitata, sacrificando parzialmente la componente d a favore della capacità di carico del motore.

2. **Slew-rate delle tensioni:** $|\Delta\mathbf{v}| \leq \Delta V_{max}$, per evitare brusche commutazioni che potrebbero danneggiare l'isolamento del motore.

A differenza della formulazione senza vincoli, dove la soluzione è ricavabile analiticamente, la presenza di disuguaglianze trasforma il problema di ottimizzazione in un problema di *Quadratic Programming* (QP), la cui risoluzione efficiente su hardware *embedded* rappresenta una sfida centrale per il controllo in tempo reale [5, 6].

3.4.1 Formulazione del problema QP

Il funzionale di costo J viene riscritto nella forma standard per solutori QP:

$$\min_{\Delta\mathbf{U}} \left(\frac{1}{2} \Delta\mathbf{U}^T \mathbf{E} \Delta\mathbf{U} + \mathbf{F}^T \Delta\mathbf{U} \right) \tag{3.21}$$

soggetto ai vincoli lineari $\mathbf{M}\Delta\mathbf{U} \leq \boldsymbol{\gamma}$. Per tradurre i limiti su $\mathbf{U}$ e $\Delta\mathbf{U}$ in questa forma, si esprime $\mathbf{U}$ in funzione di $\Delta\mathbf{U}$ tramite l'operatore di integrazione:

$$\mathbf{U} = \mathbf{U}(k-1) + \mathbf{C}\Delta\mathbf{U} \tag{3.22}$$

dove $\mathbf{C}$ è una matrice triangolare inferiore di uno. I vincoli diventano quindi:

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{C} \\ -\mathbf{C} \end{bmatrix} \Delta\mathbf{U} \leq \begin{bmatrix} \Delta\mathbf{U}_{max} \\ -\Delta\mathbf{U}_{min} \\ \mathbf{U}_{max} - \mathbf{U}(k-1) \\ -\mathbf{U}_{min} + \mathbf{U}(k-1) \end{bmatrix} \tag{3.23}$$

che corrisponde esattamente alla forma compatta $\mathbf{M}\Delta\mathbf{U} \leq \boldsymbol{\gamma}$.

3.4.2 Ottimizzazione vincolata e dualità

Per comprendere le proprietà della soluzione ottima in presenza di vincoli attivi, si definisce la funzione Lagrangiana $\mathscr{L}$:

$$\mathscr{L}(\Delta\mathbf{U}, \boldsymbol{\lambda}) = \frac{1}{2}\Delta\mathbf{U}^T\mathbf{E}\Delta\mathbf{U} + \mathbf{F}^T\Delta\mathbf{U} + \boldsymbol{\lambda}^T(\mathbf{M}\Delta\mathbf{U} - \boldsymbol{\gamma}) \tag{3.24}$$

dove $\boldsymbol{\lambda} \geq 0$ rappresenta il vettore dei moltiplicatori di Lagrange. La relazione tra il valore dei moltiplicatori e lo stato dei vincoli è immediata:

- Quando un vincolo non è attivo ($\lambda_i = 0$), la condizione di ottimalità si riduce al caso non vincolato.
- Quando un vincolo è attivo (es. saturazione di tensione), il moltiplicatore $\lambda_i > 0$ agisce come una ”forza correttiva” che mantiene la soluzione sulla frontiera della regione ammissibile.

La risoluzione del problema primale in tempo reale può risultare computazionalmente onerosa, specialmente all'aumentare dell'orizzonte di predizione. La minimizzazione di $\mathscr{L}$ rispetto a $\Delta\mathbf{U}$ (assumendo il vincolo non attivo) fornisce la soluzione in forma chiusa $\Delta\mathbf{U} = -\mathbf{E}^{-1}(\mathbf{F} + \mathbf{M}^T\boldsymbol{\lambda})$. Sostituendo tale espressione nella Lagrangiana, si perviene al *problema duale*:

$$\min_{\boldsymbol{\lambda}\geq 0} \left(\frac{1}{2}\boldsymbol{\lambda}^T\mathbf{H}\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\mathbf{K} + \frac{1}{2}\boldsymbol{\gamma}^T\mathbf{E}^{-1}\boldsymbol{\gamma} \right) \tag{3.25}$$

dove le matrici $\mathbf{H}$ e $\mathbf{K}$ sono definite come:

$$\mathbf{H} = \mathbf{M}\mathbf{E}^{-1}\mathbf{M}^T \tag{3.26}$$

$$\mathbf{K} = \boldsymbol{\gamma} + \mathbf{M}\mathbf{E}^{-1}\mathbf{F} \tag{3.27}$$

Il passaggio al dominio duale trasforma radicalmente il compito di ottimizzazione: l'obiettivo non è più la ricerca diretta del vettore di comando $\Delta\mathbf{U}$, bensì la determinazione del vettore di moltiplicatori $\boldsymbol{\lambda} \geq 0$ che identifica quali vincoli siano attivi al minimo.

Il problema duale presenta un vantaggio strategico: la matrice $\mathbf{H}_{dual} = \mathbf{M}\mathbf{E}^{-1}\mathbf{M}^T$ ha dimensione pari al numero di vincoli, tipicamente inferiore al numero di variabili di decisione del problema primale. Questa riduzione di dimensionalità rende il problema duale ideale per l'implementazione su microcontrollore.

La risoluzione di questo problema di ottimizzazione per $\boldsymbol{\lambda}$ è il cuore algoritmico che andremo a implementare. Nello specifico, utilizzeremo l'algoritmo di ottimizzazione quadratica di *Hildreth*, il quale permette di risolvere il problema duale in modo iterativo, garantendo una convergenza robusta alla soluzione ottima. Tale approccio trasforma l'onere del calcolo in un processo di aggiornamento sequenziale dei moltiplicatori.

3.4.3 L'algoritmo di ottimizzazione di Hildreth

Il problema duale di ottimizzazione, formulato nel paragrafo precedente, richiede la determinazione del vettore di moltiplicatori $\boldsymbol{\lambda} \geq 0$ che minimizza la funzione quadratica [7, 8]:

$$\min_{\boldsymbol{\lambda} \geq 0} \left(\frac{1}{2} \boldsymbol{\lambda}^T \mathbf{H} \boldsymbol{\lambda} + \boldsymbol{\lambda}^T \mathbf{K} \right) \tag{3.28}$$

dove $\mathbf{H} = \mathbf{M}\mathbf{E}^{-1}\mathbf{M}^T$ è una matrice simmetrica e definita positiva. Per risolvere questo problema di ottimizzazione vincolata in tempo reale, si utilizza il metodo iterativo di *Hildreth*.

L'algoritmo aggiorna sequenzialmente ogni componente del vettore $\boldsymbol{\lambda}$, garantendo che ad ogni passo il vincolo $\lambda_i \geq 0$ venga rispettato. Per ogni iterazione p e per ogni moltiplicatore λ_i, il nuovo valore è calcolato come:

$$\lambda_i^{(p+1)} = \max \left(0, \lambda_i^{(p)} - \frac{1}{H_{ii}} \left(K_i + \sum_{j=1}^{i-1} H_{ij} \lambda_j^{(p+1)} + \sum_{j=i}^{m} H_{ij} \lambda_j^{(p)} \right) \right) \tag{3.29}$$

dove H_{ij} sono gli elementi della matrice $\mathbf{H}$ e m è il numero totale di vincoli.

Il metodo di Hildreth risulta particolarmente efficace per l'implementazione sul sistema di controllo del motore PMSM per diverse ragioni:

- **Semplicità algoritmica:** Non richiede inversioni di matrici complesse durante il ciclo di controllo; l'operazione dominante è una somma pesata, facilmente implementabile in aritmetica a virgola mobile.

- **Convergenza:** Poiché la matrice $\mathbf{H}$ è definita positiva, l'algoritmo garantisce la convergenza alla soluzione ottima in un numero limitato di iterazioni, a patto che il sistema sia ben condizionato.

- **Determinismo:** Il numero di operazioni per iterazione è costante, il che rende il tempo di esecuzione dell'algoritmo prevedibile, parametro critico per garantire la stabilità del loop.

In fase di implementazione, si impone un numero massimo di iterazioni p_{max} per assicurare che, anche in condizioni di saturazione estrema dei vincoli, il microcontrollore sia sempre in grado di fornire un comando di tensione entro la fine dell'intervallo di campionamento. Tale soglia di iterazioni, calibrata in base al tempo di calcolo disponibile, permette di bilanciare la precisione dell'ottimizzazione con la rigidità temporale richiesta dal controllo degli azionamenti.

3.5 Funzioni di Laguerre

Un limite intrinseco degli algoritmi MPC classici risiede nella crescita lineare del numero di variabili di decisione rispetto all'orizzonte di controllo N_c. Per orizzonti di predizione molto estesi, necessari in sistemi con dinamiche lente o per garantire stabilità a lungo termine, il problema di ottimizzazione risultante può diventare proibitivo per le risorse di calcolo limitate di un microcontrollore industriale.

Originariamente introdotte nell'area dell'identificazione dei sistemi per rappresentare la risposta all'impulso di sistemi dinamici stabili [9], le funzioni di Laguerre sono state successivamente adattate per la parametrizzazione della traiettoria di controllo nell'MPC [1]. L'idea fondamentale consiste nel descrivere la variazione della traiettoria di controllo $\Delta u(k_i+k)$ non più come una sequenza di N_c singoli comandi, ma come una combinazione lineare di un numero ridotto (N) di funzioni base:

$$\Delta u(k_i+k) = \sum_{j=1}^{N} c_j(k_i) l_j(k) = \mathbf{L}(k)^T \boldsymbol{\eta} \tag{3.30}$$

dove:

- k_i è l'istante iniziale della finestra mobile;
- k è l'istante di campionamento futuro;
- $l_j(k)$ sono le funzioni base di Laguerre;
- $\boldsymbol{\eta} = [c_1, c_2, \ldots, c_N]^T$ è il vettore dei coefficienti di Laguerre, che costituisce il nuovo vettore delle variabili di decisione per l'ottimizzazione;
- $\mathbf{L}(k) = [l_1(k), l_2(k), \ldots, l_N(k)]^T$ è il vettore delle funzioni base all'istante k.

La funzione base ℓ_j è definita attraverso la trasformata Z e dipende essenzialmente da due parametri di progetto: il polo di Laguerre α ($0 \leq \alpha < 1$) e il numero di campioni N che definisce l'orizzonte delle funzioni.

La funzione di base di Laguerre al passo k è data da [1]:

$$L_1(k) = \sqrt{1-\alpha^2}\alpha^{k-1} \tag{3.31}$$

$$L_j(k) = \alpha L_j(k-1) + (1-\alpha^2)\alpha^{k-2}\sum_{i=1}^{j-1}(-\alpha)^{j-i-1}L_i(1) \tag{3.32}$$

- **Polo di Laguerre (α):** Un valore di α vicino a 0 rende le funzioni base molto rapide (ideali per sistemi con dinamiche veloci), mentre un α vicino a 1 rende le funzioni più "lente", permettendo di catturare dinamiche con costanti di tempo elevate. Per il controllo corrente di un motore PMSM, α viene solitamente scelto in funzione del tempo di campionamento T_s e della costante di tempo elettrica τ_e.
- **Orizzonte delle funzioni (N):** Rappresenta il numero di campioni di riferimento su cui le funzioni base sono definite. Esso definisce la "durata" della base di funzioni.

L'orizzonte di controllo N_c tradizionale viene sostituito dal numero di termini N utilizzato per espandere la traiettoria, in congiunzione con il parametro a (polo di Laguerre). Il parametro a definisce il tasso di decadimento delle funzioni base: un valore di a più elevato permette di ottenere un orizzonte di controllo esteso utilizzando un numero ridotto di parametri N nel processo di ottimizzazione.

Si osserva che per $a = 0$ e $N = N_c$, il modello si riconduce al caso MPC tradizionale, in cui ogni singolo comando futuro viene ottimizzato indipendentemente. La parametrizzazione tramite funzioni di Laguerre permette quindi di gestire la complessità della traiettoria di controllo in modo compatto, riducendo drasticamente il numero di gradi di libertà del problema QP senza rinunciare alla capacità di modellare dinamiche di controllo a lungo termine. Questo approccio trasforma il problema di controllo in un'ottimizzazione del vettore dei coefficienti $\mathbf{c}$, mantenendo intatta la capacità di predizione sull'intero orizzonte N_p ma operando in uno spazio di ricerca ridotto. Tale tecnica si rivela decisiva per implementare algoritmi predittivi su piattaforme embedded, dove il bilanciamento tra performance di controllo e disponibilità ciclica di CPU rappresenta la sfida progettuale primaria.

3.5.1 Funzionale di costo parametrizzato

La parametrizzazione tramite funzioni di Laguerre consente di esprimere il funzionale di costo J direttamente in termini del vettore dei coefficienti $\boldsymbol{\eta}$. Seguendo l'approccio classico proposto in [1], il funzionale di costo può essere formulato basandosi sull'errore di inseguimento tra il segnale di riferimento $r(k_i)$ e l'uscita predetta $y(k_i+m|k_i)$:

$$J = \sum_{m=1}^{N_p}\left(r(k_i) - y(k_i+m|k_i)\right)^T\left(r(k_i) - y(k_i+m|k_i)\right) + \boldsymbol{\eta}^T\mathbf{R}_L\boldsymbol{\eta} \tag{3.33}$$

dove $\mathbf{R}_L$ è una matrice diagonale che penalizza l'ampiezza dei coefficienti di Laguerre. Sfruttando le proprietà di ortonormalità delle funzioni base, l'energia del segnale di controllo viene preservata nella trasformazione, permettendo di riformulare il costo in modo analogo alla teoria del controllo lineare quadratico discreto (DLQR):

$$J = \sum_{m=1}^{N_p} x(k_i+m|k_i)^T \mathbf{Q} x(k_i+m|k_i) + \boldsymbol{\eta}^T \mathbf{R}_L \boldsymbol{\eta} \tag{3.34}$$

Questa formulazione è di fondamentale importanza poiché stabilisce un legame diretto tra l'MPC e il sistema DLQR. Tale connessione permette di utilizzare i consolidati risultati della teoria classica del controllo ottimo per l'analisi, il tuning e la progettazione del sistema MPC, garantendo proprietà di stabilità e prestazioni coerenti con le aspettative del controllo ottimo.

3.5.2 Minimizzazione del funzionale e connessione con il DLQR

Sostituendo la dinamica predetta $\mathbf{Y} = \mathbf{F}x(k_i) + \Phi_L \boldsymbol{\eta}$ nel funzionale di costo, il problema di minimizzazione assume la forma quadratica rispetto al vettore di coefficienti $\boldsymbol{\eta}$:

$$J = \boldsymbol{\eta}^T \Omega \boldsymbol{\eta} + 2\boldsymbol{\eta}^T \Psi x(k_i) + \sum_{m=1}^{N_p} x(k_i)^T (A^T)^m Q A^m x(k_i) \tag{3.35}$$

dove le matrici Ω e Ψ sono definite come:

$$\Omega = \sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L \tag{3.36}$$

$$\Psi = \sum_{m=1}^{N_p} \phi(m) Q A^m \tag{3.37}$$

essendo $\phi(m)$ il vettore delle funzioni base di Laguerre all'istante m. La soluzione ottima $\boldsymbol{\eta}^*$, ottenuta annullando la derivata parziale $\partial J/\partial \boldsymbol{\eta} = 0$, è data da:

$$\boldsymbol{\eta}^* = -\Omega^{-1} \Psi x(k_i) \tag{3.38}$$

3.5.3 Analisi della soluzione ottima

Per determinare il valore minimo del costo J_{min}, si applica la tecnica del completamento del quadrato:

$$J = (\boldsymbol{\eta} + \Omega^{-1}\Psi x(k_i))^T \Omega (\boldsymbol{\eta} + \Omega^{-1}\Psi x(k_i)) + x(k_i)^T P_{dmpc} x(k_i) \tag{3.39}$$

dove la matrice P_{dmpc} è definita come:

$$P_{dmpc} = \sum_{m=1}^{N_p} (A^T)^m Q A^m - \Psi^T \Omega^{-1} \Psi \tag{3.40}$$

Sostituendo la soluzione ottima $\boldsymbol{\eta}^*$ nel funzionale, si ottiene:

$$J_{min} = x(k_i)^T P_{dmpc} x(k_i) \tag{3.41}$$

Questo risultato stabilisce una connessione formale tra il sistema MPC basato su Laguerre e il regolatore discreto lineare quadratico (DLQR). La matrice P_{dmpc} gioca un ruolo analogo alla soluzione dell'equazione di Riccati nel DLQR classico. Tale analogia permette di utilizzare i consolidati strumenti della teoria del controllo ottimo per la sintesi dei pesi Q e R_L, assicurando che la parametrizzazione tramite Laguerre non solo riduca l'onere computazionale, ma mantenga intatte le proprietà di stabilità e ottimalità del sistema a anello chiuso.

Questa riformulazione, oltre a semplificare la struttura computazionale riducendo le dimensioni del problema alle sole variabili $\boldsymbol{\eta}$, fornisce una solida base teorica per l'analisi della stabilità del controllore in tempo reale.

3.5.4 Legge di controllo a retroazione di stato

Una volta ottenuta la soluzione ottima $\boldsymbol{\eta}^* = -\Omega^{-1}\Psi x(k_i)$, l'azione di controllo effettivamente applicata al sistema all'istante k_i è data dalla prima variazione del comando $\Delta u(k_i)$. Sfruttando la forma vettoriale $\Delta u(k_i) = \mathbf{L}(0)^T \boldsymbol{\eta}^*$, si ottiene la legge di controllo:

$$\Delta u(k_i) = \mathbf{L}(0)^T \boldsymbol{\eta}^* = -\mathbf{L}(0)^T \Omega^{-1} \Psi x(k_i) \tag{3.42}$$

Definendo il guadagno del controllore MPC come $\mathbf{K}_{mpc} = \mathbf{L}(0)^T \Omega^{-1} \Psi$, la legge di controllo assume la forma compatta di retroazione di stato lineare:

$$\Delta u(k) = -\mathbf{K}_{mpc} x(k) \tag{3.43}$$

dove il vettore $\mathbf{L}(0)^T$ è definito in funzione dei parametri di Laguerre N e a come:

$$\mathbf{L}(0)^T = \sqrt{1-a^2}\begin{bmatrix} 1 & -a & a^2 & -a^3 & \dots & (-1)^{N-1} a^{N-1} \end{bmatrix} \tag{3.44}$$

Questa rappresentazione è di estremo valore pratico: essa rivela che, nel caso senza vincoli, l'MPC parametrizzato tramite Laguerre agisce esattamente come un controllore lineare ottimo. Tale proprietà è fondamentale per:

1. **Analisi di stabilità:** Permette di calcolare i poli del sistema a anello chiuso e verificarne la stabilità tramite il raggio spettrale di $(A - B\mathbf{K}_{mpc})$.

2. **Verifica del comportamento:** Il guadagno $\mathbf{K}_{mpc}$ può essere calcolato *off-line* e confrontato con i guadagni di un regolatore PID o LQR classico, fornendo un riferimento immediato per la validazione delle prestazioni.

In presenza di vincoli, il comportamento del sistema devia da questa legge lineare a causa della saturazione, ma la forma $\Delta u(k) = -\mathbf{K}_{mpc}x(k)$ rimane il punto di riferimento fondamentale per comprendere la direzione dell'azione di controllo.

3.6 Stabilità e robustezza del sistema

La stabilità asintotica del sistema a ciclo chiuso è un requisito fondamentale per l'affidabilità di qualsiasi controllore industriale. Nel contesto del controllo predittivo basato su modello (MPC), l'analisi della stabilità presenta sfide peculiari: quando i vincoli operativi (saturazione dell'inverter, limiti di corrente) vengono attivati, la legge di controllo cessa di essere lineare e il problema assume una natura intrinsecamente non lineare [1].

A differenza dei sistemi lineari tempo-invarianti, per i quali la stabilità è garantita dalla posizione dei poli nel cerchio unitario, nel caso dell'MPC occorre dimostrare che l'ottimizzazione del funzionale di costo J agisca come una *funzione di Lyapunov* decrescente. In altre parole, l'algoritmo deve garantire che, ad ogni istante di campionamento, il sistema spenda energia per ridurre l'errore verso il riferimento, assicurando che lo stato converga all'origine. Di seguito vengono analizzate le due strategie principali per garantire tale proprietà: la prima basata su vincoli severi sullo stato terminale e la seconda, più orientata all'applicazione pratica, basata sull'estensione dell'orizzonte di predizione.

3.6.1 Stabilità con vincoli sullo stato terminale

Il principio del *receding horizon* garantisce che, ad ogni istante k_i, l'ottimizzazione del funzionale J avvenga nel rispetto dei vincoli. Una condizione sufficiente per la stabilità asintotica consiste nell'imporre un vincolo di uguaglianza sullo stato finale dell'orizzonte di predizione:

$$x(k_i + N_p|k_i) = 0 \tag{3.45}$$

In questo modo, il valore minimo del funzionale di costo J_{min} all'istante k_i funge da funzione di Lyapunov $V(x(k_i), k_i)$. Il libro [1] dimostra che, sotto l'ipotesi che il problema di ottimizzazione ammetta sempre soluzione, la funzione $V(x(k_i), k_i)$ risulta strettamente

decrescente:

$$V(x(k_i+1),k_i+1)-V(x(k_i),k_i) \leq -x(k_i)^T Qx(k_i) - \Delta u(k_i)^T R\Delta u(k_i) < 0 \tag{3.46}$$

Questa disuguaglianza assicura che l'energia del sistema tenda asintoticamente a zero, garantendo la stabilità del sistema a ciclo chiuso [3, 10].

3.6.2 Stabilità con orizzonte di predizione esteso

Nella pratica industriale, il vincolo severo $x(k_i+N_p|k_i)=0$ risulta spesso troppo restrittivo, poiché può portare a una regione di ammissibilità vuota (il problema QP non trova soluzione). Si preferisce quindi una formulazione più flessibile [11, 12]:

- **Orizzonte ampio:** Si utilizza un orizzonte di predizione N_p sufficientemente lungo da coprire il transitorio del sistema.
- **Convergenza al DLQR:** Per N_p grande, la traiettoria di controllo ottima converge verso quella dettata da un regolatore lineare quadratico discreto (DLQR) con le medesime matrici di peso Q e R.

Quando il sistema opera in condizioni in cui l'orizzonte N_p è ampio, si verifica che il termine terminale $x(k_i+N_p|k_i)^T Qx(k_i+N_p|k_i) \approx 0$, poiché il DLQR sottostante è intrinsecamente stabile. In tale configurazione, la stabilità è garantita dalla capacità del controllore di guidare lo stato all'interno di un *terminal constraint set* X_0 che contiene l'origine.

È importante notare che, in presenza di un modello interno includente un integratore, l'aumento dell'orizzonte di predizione N_p può tuttavia introdurre problemi di condizionamento numerico della matrice di ottimizzazione. In letteratura, tale problematica viene tradizionalmente mitigata mediante l'uso di retroazioni di stato ausiliarie o tramite il tuning mirato dei parametri di orizzonte [13, 14]. Una volta raggiunto il set X_0, il sistema può essere stabilizzato da una legge di controllo lineare convenzionale (approccio *dual-mode*), eliminando la necessità di vincoli di uguaglianza terminali severi e migliorando significativamente la robustezza numerica dell'ottimizzazione [1].

Queste considerazioni teoriche costituiscono il fondamento del capitolo successivo, in cui verrà illustrata la metodologia di *Model-Based Design* adottata. Si descriverà il processo di traduzione del modello matematico in un firmware esecutivo, garantendo l'integrazione tra la logica di controllo sviluppata in ambiente Simulink e l'architettura di calcolo deterministica del microcontrollore.

Capitolo 4

Implementazione dell'algoritmo di controllo nell'ambiente di sviluppo

L'architettura di riferimento per il controllo di un motore PMSM è costituita dal classico schema *Field Oriented Control* (FOC), basato su una struttura gerarchica a due anelli: un anello esterno di velocità e un anello interno di regolazione delle correnti. In tale configurazione, l'anello esterno elabora l'errore tra la velocità di riferimento ω_{ref} e quella misurata, generando un riferimento di coppia τ_{ref} che viene convertito in riferimenti di corrente $(i_{d,ref}, i_{q,ref})$ [15].

L'interesse per l'MPC applicato ai drive elettrici rimane elevato, come confermato dalle recenti evoluzioni nelle tecniche di controllo robusto e nell'efficienza computazionale [16]. Per questo il lavoro di tesi si concentra sull'ottimizzazione dell'anello interno, dove il tradizionale controllore PI viene sostituito da un'architettura *Model Predictive Torque Control* (MP-TC) [15]. Mentre il loop di velocità rimane affidato a un regolatore PI standard — sufficientemente dinamico per gestire la parte meccanica, intrinsecamente più lenta — l'anello di corrente interno viene interamente riprogettato mediante MPC per gestire direttamente le grandezze elettromagnetiche [15]. Questa scelta è supportata dalle seguenti motivazioni tecniche:

1. **Separazione delle dinamiche:** La parte meccanica presenta una costante di tempo superiore a quella elettrica; la sostituzione del loop interno consente di incrementare la larghezza di banda del sistema.

2. **Gestione nativa dei vincoli:** A differenza del PI, l'MPC integra le limitazioni di tensione e corrente direttamente nel problema di ottimizzazione, garantendo che i comandi rimangano sempre entro la regione ammissibile.

3. **Linearizzazione per modelli predittivi:** Il modello bilineare del motore, disaccoppiato in modelli lineari, permette di utilizzare le tecniche di predizione MPC per il controllo indipendente degli assi d e q [15].

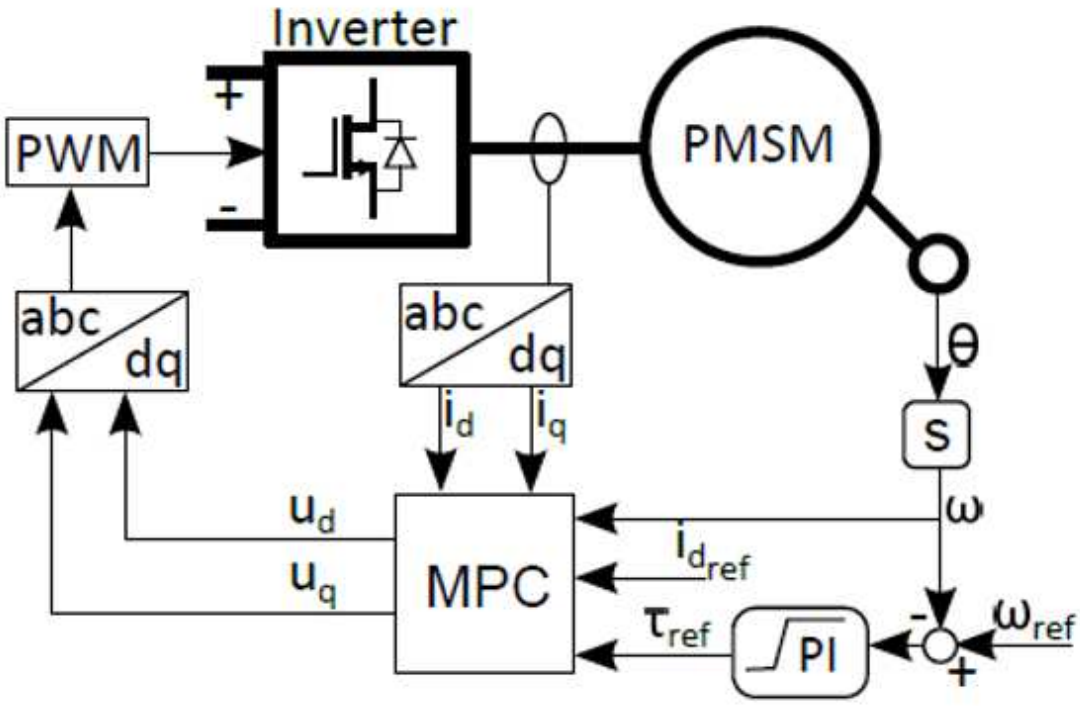


Figura 4.1: Architettura gerarchica: anello esterno di velocità e anello interno basato su MPC.

La funzione primaria del blocco MPC è l'elaborazione delle correnti misurate (i_d, i_q) e dei relativi riferimenti per il calcolo dei comandi di tensione ottimali (u_d, u_q). Come illustrato in Figura 4.1, l'architettura segue una precisa pipeline:

1. **Input:** Acquisizione delle correnti di fase e del riferimento dall'anello esterno.
2. **Elaborazione MPC:** Calcolo delle tensioni ottime per guidare il sistema lungo la traiettoria desiderata, rispettando i vincoli di saturazione.
3. **Attuazione:** Invio dei segnali al blocco di *Output Scaling* e modulazione PWM per il pilotaggio del PMSM [15].

4.1 Validazione in Ambiente di Simulazione

Per garantire una transizione affidabile tra la teoria e l'implementazione pratica, l'algoritmo è stato validato in un ambiente di simulazione ad alta fedeltà. L'ambiente MATLAB/Simulink è stato configurato per rispecchiare fedelmente l'architettura hardware del banco prova. È stata adottata la libreria *Motor Control Blockset* (MCB), configurando il modello del motore PMSM tramite la routine di inizializzazione:

```
pmsm_motor1 = mcb_SetPMSMMotorParameters('Teknic2310P');
```
(4.1)

Tale approccio permette di importare i parametri elettromeccanici reali, garantendo che la dinamica simulata sia rappresentativa del comportamento del motore Teknic. La simulazione include inoltre una modellazione dettagliata delle limitazioni dell'inverter, operando con il medesimo tempo di campionamento ($T_s = 25\,\mu s$) previsto per l'implementazione sul microcontrollore. Tale impostazione permette di validare non solo il tracking dei riferimenti, ma anche la stabilità numerica del solutore in condizioni di carico variabili, preparando il terreno per la successiva fase di implementazione su hardware dedicato descritta nel Capitolo 5.

4.2 Traduzione del modello teorico in ambiente di calcolo

Come analizzato nel Capitolo 3, l'efficienza computazionale dell'MPC risiede nella netta distinzione tra le operazioni pesanti eseguite *off-line* e quelle leggere eseguite in tempo reale dal microcontrollore. Il codice MATLAB non funge da mero strumento di simulazione, ma agisce come un vero e proprio *pre-processore strutturale*. Nello specifico, l'ambiente MATLAB viene impiegato per:

1. **Discretizzazione e costruzione del modello esteso:** Conversione del sistema continuo in discreto e costruzione dello stato aumentato per la reiezione dei disturbi, garantendo l'integrazione del modello nel tempo.

2. **Selezione e tuning dei parametri MPC:** Configurazione dell'orizzonte di predizione (N_p), dell'orizzonte di controllo (N_c) e dei parametri delle funzioni di Laguerre (a, N). Questa fase definisce la complessità del problema di ottimizzazione, bilanciando accuratamente la dinamica del controllo con le risorse computazionali del processore TI.

3. **Configurazione dinamica dei vincoli:** Calcolo della matrice A_{cons} che definisce il politopo di ammissibilità dei comandi. Questa matrice permette al risolutore di mappare istantaneamente i limiti di tensione e corrente dello stadio di potenza (inverter) direttamente nel dominio di controllo.

4. **Ottimizzazione del funzionale di costo:** Calcolo preventivo delle matrici di costo (Ω e Ψ) e pre-inversione della matrice Hessiana (Ω^{-1}). Questa operazione è cruciale poiché elimina la necessità di eseguire inversioni di matrice, computazionalmente proibitive, durante ogni ciclo di campionamento.

5. **Definizione delle tabelle di Gain Scheduling:** Calcolo delle matrici ottime su una griglia di punti operativi (velocità e resistenza), garantendo che il controllore mantenga prestazioni ottimali anche al variare delle condizioni di lavoro del motore.

Tale approccio trasforma il problema di ottimizzazione in una sequenza di operazioni lineari semplici (moltiplicazioni matrice-vettore), trasformando un algoritmo teoricamente complesso in un firmware *real-time* deterministico, pronto per essere eseguito dal processore.

4.2.1 Discretizzazione e costruzione del modello esteso

Il punto di partenza dell'implementazione è la discretizzazione del sistema continuo nel dominio z. Definendo il tempo di campionamento T_s, otteniamo le matrici del modello discreto A_d e B_d che descrivono l'evoluzione del sistema tra due campioni successivi.

L'estensione del modello per il controllo della variazione di comando Δu — necessaria per eliminare l'errore a regime e conferire robustezza al controllore — viene effettuata costruendo il modello di stato aumentato:

$$\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} \Delta u(k) \tag{4.2}$$

Il codice MATLAB riportato di seguito implementa questa estensione, creando la struttura a blocchi necessaria per il calcolo delle matrici di predizione:

Listing 4.1: Costruzione delle matrici del modello discreto ed esteso.

```
% Costruzione delle matrici del modello discreto
Ad = [1 - (curr_Rs*Ts/Ld),        curr_we*Ts*Lq/Ld;
        -curr_we*Ts*Ld/Lq,    1 - (curr_Rs*Ts/Lq)];
Bd = [Ts/Ld, 0; 0,  Ts/Lq];
Cd = eye(2);

% Creazione del modello esteso (A_e, B_e, C_e)
A_e = [Ad, zeros(2,2); Cd*Ad, eye(2)];
B_e = [Bd; Cd*Bd];
C_e = [zeros(2, 2), eye(2)];
```

Questa fase rispecchia fedelmente le equazioni del Capitolo 3.1.1 relative alla costruzione del vettore di stato aumentato.

4.2.2 Selezione dei parametri di controllo e delle funzioni di Laguerre

La fase di *tuning* dei parametri dell'MPC gioca un ruolo cruciale nel determinare la dinamica del sistema a ciclo chiuso. La selezione dei valori è stata guidata dalla necessità di bilanciare la prontezza di risposta del regolatore con i vincoli di stabilità numerica e i limiti computazionali imposti dall'architettura *real-time* del microcontrollore.

I parametri adottati per l'implementazione sono definiti nel codice seguente:

Listing 4.2: Definizione dei parametri di controllo e pesi di ottimizzazione.

```
a  = [0.8; 0.8];     % Polo delle funzioni di Laguerre
N  = [1; 1];         % Numero di funzioni base per asse
Np = 50;             % Orizzonte di predizione
Nc = 4;              % Orizzonte di vincolo

% Tuning pesi (Q=identita', R=diag)
Q = C_e' * C_e;
R = diag([5, 5]);
```

Parametrizzazione tramite Laguerre

L'utilizzo delle funzioni di Laguerre permette di descrivere il segnale di controllo futuro come una combinazione lineare di funzioni base, riducendo significativamente il numero di variabili di decisione. La scelta di $N = [1;1]$ è il risultato di un'analisi di sensibilità del solutore: sebbene gradi di libertà superiori ($N > 1$) offrano una maggiore flessibilità teorica, essi introducono instabilità numeriche dovute alla maggiore complessità computazionale e alla sensibilità del solutore di Hildreth alle incertezze del modello. Il primo modo di Laguerre è risultato sufficiente a catturare la dinamica transitoria dominante del PMSM.

Il parametro di scala $a = 0.8$ definisce il polo delle funzioni base. Tale valore è stato selezionato per garantire un'elevata reattività del controllore nelle fasi iniziali del transitorio di corrente, ottimizzando la risposta a variazioni repentine del riferimento.

Orizzonte di predizione e gestione dei vincoli

L'orizzonte di predizione $N_p = 50$ è stato calibrato per coprire la dinamica elettrica dominante del motore. Con un tempo di campionamento $T_s = 25\,\mu s$, una finestra di 50 campioni corrisponde a $1.25\,ms$, valore coerente con la costante di tempo elettrica del sistema, consentendo all'MPC di pianificare correttamente la traiettoria di corrente durante il transitorio di salita.

Il parametro $N_c = 4$ è stato configurato per definire l'orizzonte su cui viene imposto il rispetto dei vincoli di tensione. Attraverso analisi empiriche, si è riscontrato che questo valore garantisce una pianificazione ottimale della traiettoria di tensione, evitando la saturazione improvvisa dell'inverter. È stato verificato che configurazioni con orizzonti troppo estesi (es. $N_p > 100$) o gradi di libertà $N > 1$ compromettono la stabilità del sistema a causa dell'accumulo di errori di modellizzazione e dell'eccessivo onere computazionale, che impedisce la convergenza del solutore entro il limite di $25\,\mu s$.

Pesi del funzionale di costo

La matrice $Q = C_e^T C_e$ assegna un peso unitario allo stato aumentato, focalizzando l'ottimizzazione sull'azzeramento dell'errore di corrente. La matrice di penalizzazione dei comandi $R = \text{diag}(5,5)$ è stata tarata per mitigare l'aggressività dell'azione di controllo. Un peso maggiore su R ha permesso di ridurre il *chattering* del segnale di comando, proteggendo gli switch dell'inverter da sollecitazioni eccessive e garantendo una risposta priva di sovraelongazioni (*overshoot*) indesiderate.

La configurazione finale rappresenta il giusto compromesso tra prestazioni dinamiche, robustezza numerica e compatibilità con l'architettura di calcolo, consentendo un'esecuzione deterministica dell'algoritmo di controllo.

4.2.3 Configurazione dinamica dei vincoli

Uno degli aspetti peculiari dell'MPC è la capacità di gestire nativamente le limitazioni fisiche del sistema. Nel nostro caso, i vincoli riguardano sia l'ampiezza delle tensioni di uscita (u_d, u_q), che le variazioni ammissibili dei comandi (Δu), necessarie per evitare bruschi transitori di tensione.

Il problema di vincolo può essere formulato nella forma compatta:

$$A_{cons}\boldsymbol{\eta} \leq \mathbf{b} \tag{4.3}$$

dove **b** rappresenta il vettore dei limiti (saturazioni). Come vedremo nel Capitolo 5, il termine **b** verrà aggiornato in tempo reale ad ogni ciclo di campionamento per tenere conto dello stato corrente e della forza contro-elettromotrice (FCEM) del motore, mentre la matrice A_{cons} è calcolata *offline* tramite MATLAB.

Costruzione delle matrici di vincolo: Mu e Mdu

La costruzione di A_{cons} si basa su due funzioni fondamentali, `Mu` e `Mdu`, che proiettano i limiti temporali nel dominio dei coefficienti di Laguerre ($\boldsymbol{\eta}$):

- `Mdu`**:** Calcola la matrice di proiezione per i vincoli sulla *variazione* del comando (Δu). Essa mappa direttamente l'azione di controllo all'istante iniziale in funzione dei parametri N e a.
- `Mu`**:** Calcola la matrice per i vincoli sul *valore assoluto* del comando (u). Essendo il comando attuale la somma delle variazioni precedenti ($\sum \Delta u$), questa funzione integra nel tempo le matrici di proiezione.

L'implementazione delle matrici segue questa logica:

Listing 4.3: Calcolo offline delle matrici di vincolo per il risolutore QP.

```
% Generazione delle matrici di proiezione basate su Laguerre
M_cons  = Mu(a, N, 2, Nc);   % Vincoli su U (ampiezza)
M_delta = Mdu(a, N, 2, Nc);  % Vincoli su DeltaU (slew rate)

% Costruzione della matrice dei vincoli A_cons (politopo)
A_cons = [ M_cons; -M_cons; M_delta; -M_delta ];
```

Significato delle funzioni

Entrambe le funzioni impiegano `lagdN` per generare le matrici di transizione delle funzioni di Laguerre (A_l e L_0). La funzione `Mdu` costruisce una struttura in cui ogni riga rappresenta la proiezione della traiettoria di controllo istantanea, permettendo di imporre il vincolo $|\Delta u| \leq \Delta u_{max}$. Al contrario, `Mu` accumula l'effetto integrale delle funzioni di Laguerre, permettendo al risolutore di conoscere l'ampiezza totale di tensione richiesta al motore e di imporre $|u| \leq u_{limit}$. Questa formulazione trasforma il limite fisico del sistema in un politopo di ammissibilità nello spazio dei coefficienti $\boldsymbol{\eta}$. In tempo reale, il vettore $\mathbf{b}$ verrà aggiornato sottraendo al limite massimo di tensione applicabile il valore del comando applicato al passo precedente (u_{old}), riconducendo il tutto alla forma canonica del problema di programmazione quadratica (QP), che verrà risolto in tempo reale mediante l'algoritmo di Hildreth.

4.2.4 Ottimizzazione offline e preparazione del solutore

Come dimostrato nel Capitolo 3.5.2, la minimizzazione del funzionale di costo J porta alla risoluzione di un problema di programmazione quadratica (QP). Il blocco di codice principale, basato sulla funzione `dmpc`, calcola le matrici E (Hessiano) e H (gradiente) che definiscono la superficie di costo quadratica. In accordo con la teoria delle funzioni di Laguerre, queste matrici sono ottenute come:

$$E = \sum_{m=1}^{N_p} \Phi(m)^T Q \Phi(m) + R_{para}, \quad H = \sum_{m=1}^{N_p} \Phi(m)^T Q A_e^m \tag{4.4}$$

dove $\Phi(m)$ rappresenta la traiettoria predetta dallo stato aumentato in funzione dei coefficienti $\boldsymbol{\eta}$. La funzione `dmpc` esegue un ciclo iterativo per l'orizzonte N_p, cumulando i contributi della dinamica del motore pesati dalla matrice Q e penalizzando le variazioni di controllo tramite R_{para}.

Pre-inversione e strutturazione per il real-time

Per evitare l'inversione della matrice Hessiana E (che risulterebbe computazionalmente proibitiva in termini di cicli macchina su un sistema embedded), le matrici necessarie al risolutore

di Hildreth vengono pre-calcolate e salvate in strutture dati tabellate (*Look-up Tables*):

Listing 4.4: Pre-calcolo delle tabelle per il risolutore QP online.

```
% 1. Inversione dell'Hessiano (offline)
Inv_Omega_k = inv(Omega_k);

% 2. Matrici pre-moltiplicate per ridurre il calcolo online
A_inv_Omega_table(:,:,k) = A_cons * Inv_Omega_k;
P_table(:,:,k)           = A_cons * (Omega_k \ A_cons');
ope_table(:,:,k)         = Omega_k \ A_cons';
```

Il salvataggio di `A_inv_Omega`, `P` e `ope` rappresenta il cuore dell'ottimizzazione:

- **A_inv_Omega:** Mappa i vincoli direttamente nello spazio dei coefficienti η, semplificando la verifica di ammissibilità.
- **P e ope:** Queste matrici contengono pre-moltiplicati i termini necessari per l'algoritmo di Hildreth, riducendo le operazioni necessarie nel loop di controllo a semplici prodotti riga-colonna.

Questa strategia permette di trasformare un problema di ottimizzazione vincolata complesso in una sequenza di moltiplicazioni matriciali, garantendo che il firmware rispetti il determinismo temporale richiesto dal loop di corrente a $25\,\mu s$. In questo modo, il microcontrollore non deve mai invertire una matrice, ma limitarsi a un accesso in memoria (*look-up*) basato sullo stato corrente del sistema.

4.2.5 Analisi di robustezza tramite Gain Scheduling

Sebbene l'architettura MPC sia intrinsecamente robusta rispetto alle incertezze di modello, le variazioni delle dinamiche non lineari del PMSM — indotte prevalentemente dalla velocità elettrica ω_e e dalla deriva termica della resistenza di statore R_s — possono limitare l'efficacia del controllore in regimi di funzionamento estesi.

Al fine di superare i limiti tipici dei modelli a parametri statici, è stata implementata una strategia di *Gain Scheduling* che permette al sistema di adattare i guadagni del modello interno alle condizioni operative in tempo reale. Questa scelta progettuale è finalizzata alla massimizzazione della robustezza globale: l'adattamento dinamico garantisce infatti una risposta coerente e stabile su tutto il campo operativo, prevenendo quelle instabilità termiche e dinamiche che porterebbero a fenomeni di saturazione o perdita di controllo. Tale approccio trasforma il controllore da un sistema ottimizzato per il punto nominale ad un'architettura adattabile alla variabilità del processo, conferendo al sistema la necessaria affidabilità richiesta in scenari di esercizio industriale.

Costruzione delle tabelle di lookup multidimensionali

In fase di pre-calcolo, il sistema è stato analizzato su una griglia operativa che copre un range di velocità tra 400 e 2000 RPM e una variazione della resistenza statorica fino al $+60\%$ rispetto al valore nominale. Per ogni combinazione di velocità e resistenza, le matrici del modello discreto A_d e B_d sono state ricalcolate per riflettere la dinamica locale.

Il seguente codice MATLAB illustra la generazione delle matrici ottimali, memorizzate in strutture dati tridimensionali per facilitare l'accesso durante l'esecuzione *real-time*:

Listing 4.5: Generazione delle tabelle di Gain Scheduling 3D.

```
% Ciclo doppio su resistenza (j) e velocita' (k)
for j = 1:N_res
    curr_Rs = Rs_vec(j);
    for k = 1:N_points
        idx = (j-1)*N_points + k; % Indice linearizzato
        curr_we = (RPM_vec(k) * 2 * pi / 60) * p;

        % Aggiornamento dinamica locale
        Ad_k = [1 - (curr_Rs*Tin/Ld),  curr_we*Tin*Lq/Ld;
               -curr_we*Tin*Ld/Lq,     1 - (curr_Rs*Tin/Lq)];

        % Progettazione MPC e salvataggio nelle tabelle
        [Omega_k, Psi_k] = dmpc(A_e_k, B_e_k, a, N, Np, Q, R);

        % Salvataggio matrici indicizzate
        Psi_table(:,:,idx) = Psi_k;
        Inv_Omega_table(:,:,idx) = inv(Omega_k);
        A_inv_Omega_table(:,:,idx) = A_cons * inv(Omega_k);
        ...
    end
end
```

Implementazione e stabilità del sistema

Pur non riscontrando variazioni prestazionali significative, l'implementazione del *Gain Scheduling* garantisce un comportamento dinamico uniforme, riducendo la necessità di margini di conservatività nel tuning dei pesi Q e R. Tale struttura tabellata verrà richiamata in tempo reale nel modello Simulink tramite un selettore di indice basato sulla velocità (ω_e) e sulla stima di resistenza (R_s), assicurando un transitorio fluido tra i diversi regimi di rotazione. L'adozione di questo approccio garantisce che il controllore MPC conosca la fisica del motore per l'intero range di funzionamento. L'indice `idx` permette di selezionare in ogni istante le matrici ottimali, prevenendo discrepanze tra il modello interno al solutore e il comporta-

mento fisico reale del PMSM. Il confronto con altre tecniche di controllo verrà approfondito nel capitolo finale (capitolo 6).

4.3 Integrazione nel modello Simulink: il solutore real-time

Dopo aver consolidato le tabelle di *lookup* e le matrici di sistema in ambiente *offline*, il passo successivo è l'integrazione nel modello Simulink. Questo modello rappresenta l'architettura definitiva che viene caricata sul processore Texas Instruments C2000 per l'esecuzione in tempo reale. La funzione `mpc_laguerre_hildreth` agisce come un solutore QP *real-time*. A differenza della fase di pre-calcolo, qui il modello ha il compito di richiamare le matrici dinamicamente, risolvere il problema di ottimizzazione vincolata e fornire i comandi di tensione ottimali all'inverter.

Lo schema implementativo, illustrato in Figura 4.2, descrive il flusso di segnali del controllore:

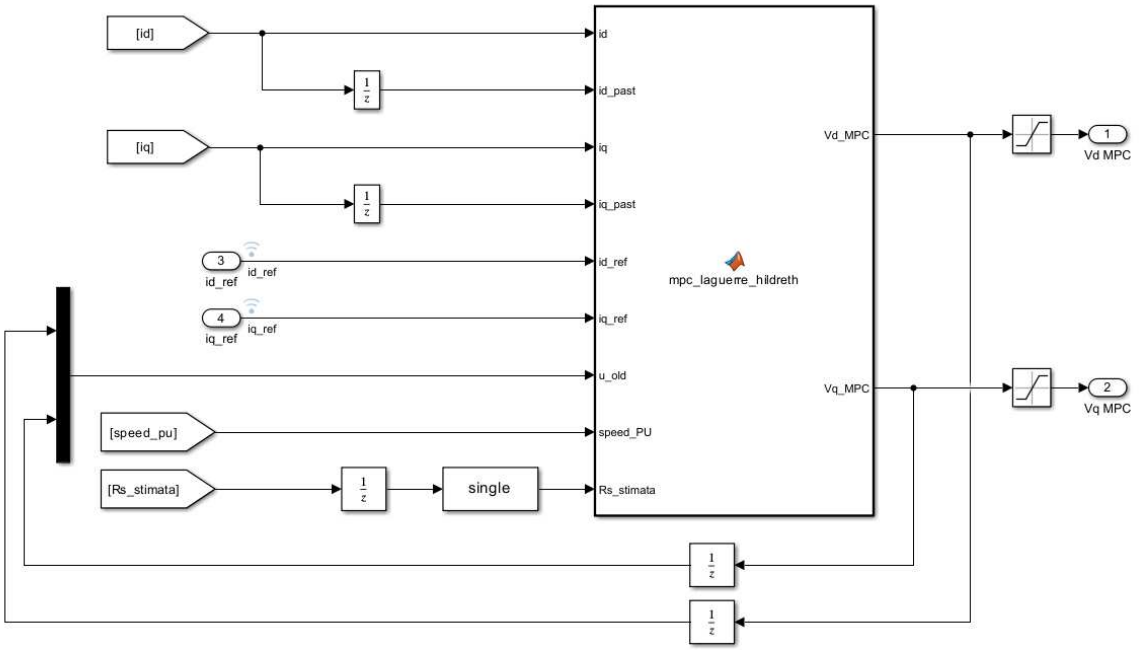


Figura 4.2: Interfaccia del blocco MPC: ingressi, uscite e logica di retroazione.

Il blocco riceve in ingresso le variabili di stato correnti e i riferimenti, processandoli secondo la seguente pipeline:

1. **Acquisizione e retroazione:** Il blocco campiona le correnti di asse d e q, insieme alle loro versioni ritardate ($i_{d,past}$, $i_{q,past}$), necessarie per ricostruire la variazione di stato (Δx) richiesta dal modello esteso.

2. **Lookup Dinamico:** In base alla velocità del rotore ($speed_PU$) e alla stima della resistenza ($Rs_stimata$), il solutore seleziona l'indice corretto nelle *Look-up Tables*

generate nella fase offline. Questo garantisce che le matrici di predizione (Ψ, Ω^{-1}, P) siano sempre coerenti con il punto di lavoro attuale del motore.

3. **Risoluzione del QP (Hildreth):** Una volta selezionate le matrici, il blocco esegue l'algoritmo di Hildreth. Il solutore calcola il vettore dei moltiplicatori di Lagrange λ che garantisce il rispetto dei vincoli ($A_{cons}\eta \leq \mathbf{b}$), minimizzando al contempo il costo quadratico.

4. **Generazione del comando:** Il risultato del processo di ottimizzazione è la variazione di comando Δu, che viene sommata al valore di tensione applicato al passo precedente (u_{old}) per generare i riferimenti finali V_d e V_q.

La struttura del blocco è progettata per essere *execution-time deterministic*: il numero di iterazioni dell'algoritmo di Hildreth è limitato e le operazioni matematiche sono ottimizzate per sfruttare l'architettura del processore TI.

4.3.1 Analisi del codice di controllo real-time

Il cuore del controllo è implementato nella funzione `mpc_laguerre_hildreth`, strutturata per garantire il determinismo temporale. Il codice si articola in tre fasi fondamentali:

Gestione delle variabili persistenti e Gain Scheduling

Il solutore utilizza variabili `persistent` per mantenere la memoria dello stato del controllore e dei filtri tra un'invocazione e l'altra. Il blocco di aggiornamento, attivato ogni 80 cicli di campionamento (pari a $2\,ms$), implementa la logica di *Gain Scheduling* basata sulla velocità e sulla resistenza statorica stimata.

Listing 4.6: Inizializzazione e logica di aggiornamento delle tabelle di lookup.

```
persistent idx_mem speed_filter Rs_filter counter_MPC;
if isempty(idx_mem)
    idx_mem = int8(1);
    speed_filter = speed_PU;
    Rs_filter = Rs;
    counter_MPC = int32(0);
end
counter_MPC = counter_MPC + int32(1);

if counter_MPC >= int32(80)
    % Filtraggio dei segnali per evitare oscillazioni dell'indice
    speed_filter = single(0.95 * speed_filter + 0.05 * speed_PU);
    Rs_filter    = single(0.95 * Rs_filter + 0.05 * Rs_stimata);

```

```
    % Calcolo dell'indice linearizzato per le tabelle 3D
    idx_speed = int8(round((speed_filter - PU_min) * PU_step_inv)) +
        int8(1);
    idx_speed = max(int8(1), min(int8(17), idx_speed));
    idx_res = int8(round((Rs_filter - Rs) / Rs_step)) + int8(1);
    idx_res = max(int8(1), min(int8(7), idx_res));

    idx_mem = (idx_res - int8(1)) * int8(17) + idx_speed;
    counter_MPC = int32(0);
end
idx_int = idx_mem;

Psi = Psi_table(:,:,idx_int);
ope = ope_table (:,:,idx_int);
A_inv_Omega = A_inv_Omega_table(:,:,idx_int);
Inv_Omega = Inv_Omega_table(:,:,idx_int);
P = P_table(:,:,idx_int);
```

La logica implementata risponde a due esigenze fondamentali per la stabilità del sistema:

1. **Filtraggio dei segnali operativi:** L'utilizzo di filtri passa-basso esponenziali ($0.95 \cdot x_{old} + 0.05 \cdot x_{new}$) su `speed_filter` e `Rs_filter` è cruciale. In assenza di questo filtraggio, il rumore di misura o lievi oscillazioni dei segnali in ingresso causerebbero una commutazione continua tra indici adiacenti delle tabelle di *lookup*. Poiché ogni indice corrisponde a una diversa matrice di controllo, tali oscillazioni produrrebbero discontinuità nel comando di tensione, compromettendo la stabilità del loop di corrente.

2. **Calcolo dell'indice linearizzato:** L'indice `idx_mem` viene calcolato linearizzando lo spazio bidimensionale (velocità, resistenza) in un unico vettore. Il motore di ricerca mappa la velocità su 17 livelli e la resistenza su 7 livelli. Le operazioni di `round` e `max/min` assicurano che il calcolo resti sempre entro i confini dei vettori pre-calcolati (clamping spaziale), evitando errori di memoria durante l'accesso alle tabelle *offline*.

Questo meccanismo di *scheduling* garantisce una transizione fluida tra i modelli lineari al variare delle condizioni operative, permettendo al sistema di operare con precisione sia in regimi di bassa velocità che in presenza di riscaldamento del motore, senza richiedere ricalcoli onerosi durante il ciclo di controllo.

Costruzione dei vincoli dinamici

In ogni istante di campionamento, il problema di ottimizzazione deve essere vincolato dalle condizioni operative correnti del motore. Poiché l'inverter ha una tensione di bus limitata,

è essenziale che il controllore conosca il margine di tensione disponibile per l'azione di controllo. Il vettore dei vincoli `b_cons` viene aggiornato dinamicamente per riflettere lo stato attuale:

Listing 4.7: Aggiornamento dinamico del vettore dei vincoli b_cons.

```
v_back_emf = single(Flux * speed_PU);
u_old_d = u_old(1);
u_old_q_active = u_old(2) - v_back_emf;

b_con = [ ud_limit - u_old_d;
        (uq_limit - v_back_emf) - u_old_q_active;
        ud_limit + u_old_d;
        (uq_limit + v_back_emf) + u_old_q_active;
        du_max; du_max;
        du_max; du_max ];

b_cons = repmat(b_con, Nc, 1);
```

Analisi della compensazione L'operazione descritta è fondamentale per la correttezza del modello predittivo:

- **Compensazione della FCEM:** La tensione di uscita dell'inverter deve bilanciare sia la caduta ohmica e induttiva, sia la **forza contro-elettromotrice** (FCEM) generata dal rotore in movimento. Sottraendo `v_back_emf` dai limiti di tensione (`uq_limit`), il controllore calcola il *margine di tensione attivo* effettivo.

- **Vincoli di Slew-Rate:** Le ultime quattro componenti del vettore `b_con` corrispondono a `du_max`, che impone un limite alla massima variazione di tensione ammissibile tra due istanti di campionamento. Questo evita commutazioni brusche che potrebbero indurre oscillazioni o correnti impulsive indesiderate.

- **Ripetizione dei vincoli (`repmat`):** L'uso di `repmat` è necessario per estendere il vincolo su tutto l'orizzonte di controllo N_c. Poiché il risolutore QP lavora sull'intero vettore $\boldsymbol{\eta}$ (che contiene i coefficienti di Laguerre per l'intero orizzonte), i limiti devono essere applicati in modo coerente per ogni grado di libertà del segnale di comando.

In sintesi, questa procedura permette all'MPC di operare conoscendo quanto margine di manovra ha a disposizione per la regolazione della coppia, evitando di richiedere tensioni non realizzabili fisicamente, che porterebbero il sistema alla saturazione incontrollata dell'inverter.

Il solutore di Hildreth

L'algoritmo del controllore si basa sul risolutore di Hildreth, una procedura iterativa duale che permette di risolvere il problema di ottimizzazione vincolata minimizzando il funzionale di costo J all'interno del politopo definito dai vincoli operativi.

L'implementazione *real-time* dell'algoritmo è riportata nel Listato 4.8:

Listing 4.8: Implementazione real-time dell'algoritmo di Hildreth.

```
% 1. Soluzione senza vincoli (Unconstrained)
dx = [id - id_past; iq - iq_past];
error_val = [id - id_ref; iq - iq_ref];
x_aug = [dx; error_val];
f_cost = single(Psi * x_aug);
eta = single(-Inv_Omega * f_cost);

% 2. Verifica violazione vincoli
A_eta = single(A_cons * eta);
kk = 0;
lambda = zeros(n_cons, 1, 'single');
for i = 1:n_cons
    if (A_eta(i) > b_cons(i))
        kk = kk + 1;
    end
end

% 3. Ottimizzazione vincolata (se kk > 0)
if (kk > 0)
    d = single(A_inv_Omega * f_cost) + b_cons;
    for km = 1:15
        lambda_p = lambda;
        for i = 1:n_cons
            w = single(P(i,:) * lambda - P(i,i) * lambda(i));
            la = single(-(w + d(i)) / P(i,i));
            lambda(i) = max(single(0), la);
        end
        % Break se la convergenza e' raggiunta
        if (single((lambda - lambda_p)' * (lambda - lambda_p)) < 1e-7)
            break;
        end
    end
    eta = eta - single(ope * lambda);
end
```

La logica di risoluzione si sviluppa in tre fasi fondamentali:

1. **Analisi della soluzione non vincolata:** Il sistema calcola inizialmente la soluzio-

ne ideale $\eta_{unc} = -\Omega^{-1} f_{cost}$, minimizzante del costo in spazio illimitato. Il termine `A_eta > b_cons` verifica il rispetto dei limiti fisici ($|u| \leq u_{limit}$ e $|\Delta u| \leq \Delta u_{max}$). Se la condizione è soddisfatta, il solutore termina istantaneamente, garantendo la massima efficienza computazionale in condizioni di regime.

2. **Proiezione dei moltiplicatori di Lagrange:** In caso di violazione, l'algoritmo attiva un ciclo iterativo per il calcolo dei moltiplicatori di Lagrange λ. Ogni iterazione proietta la soluzione all'interno del politopo ammissibile. Il numero di iterazioni è limitato ($km = 15$) per garantire il **determinismo temporale**, essenziale per la *deadline* di $25\,\mu s$.

3. **Correzione finale e attuazione:** La variazione di comando η viene corretta sottraendo il termine di proiezione $ope \cdot \lambda$, spostando la soluzione nel punto ammissibile più vicino. Il comando viene infine trasformato in tensione reale:

$$V_{d,MPC} = u_{old,d} + L0_d \cdot \eta_1, \quad V_{q,MPC} = u_{old,q,active} + L0_q \cdot \eta_2 + e_{back_emf} \tag{4.5}$$

Tale implementazione chiude il loop di controllo: l'MPC pianifica la tensione ottimale, l'algoritmo di Hildreth ne garantisce la realizzabilità fisica e la compensazione della forza contro-elettromotrice assicura che il sistema operi correttamente nel dominio del tempo reale, rispettando i vincoli dell'inverter.

4.4 Stima online della resistenza statorica

Per garantire la precisione del modello interno all'MPC anche in presenza di variazioni termiche, è stato implementato uno stimatore basato su *Sliding Mode Observer (SMO)*. Nell'architettura di controllo proposta, il sistema di *Gain Scheduling* dipende dalla conoscenza esatta di due parametri chiave: la velocità del rotore (ω_e) e la resistenza statorica (R_s).

- **Velocità (ω_e):** La disponibilità di una misura sensoristica ad alta risoluzione (es. encoder) permette di conoscere la velocità in ogni istante con precisione elevata. Di conseguenza, il modello interno all'MPC può essere aggiornato utilizzando il valore misurato, eliminando ogni incertezza su questo parametro.

- **Resistenza statorica (R_s):** Al contrario, la resistenza di statore non è una variabile misurabile direttamente in tempo reale. Sebbene le variazioni termiche degli avvolgimenti siano fenomeni relativamente lenti, la loro corretta identificazione rappresenta una *best practice* nel controllo di motori ad alte prestazioni. Inserire un meccanismo

di stima permette di mantenere il modello interno dell'MPC costantemente allineato con la realtà fisica del motore, eliminando la necessità di margini di conservatività nel tuning dei pesi e garantendo una precisione di controllo costante nel tempo, indipendentemente dalle condizioni di esercizio.

4.4.1 Stima dei parametri e osservatori di stato

L'efficacia dell'MPC, come delineato nel Capitolo 3, è intrinsecamente legata all'accuratezza del modello matematico del motore. In condizioni operative reali, parametri quali la resistenza di statore (R_s) sono soggetti a variazioni significative a causa della deriva termica degli avvolgimenti. Tali scostamenti tra modello nominale e realtà fisica — definiti come *model mismatch* — possono compromettere la precisione del tracking e la stabilità del sistema a ciclo chiuso.

Per superare questa limitazione senza ricorrere a sensoristica aggiuntiva, si è fatto ricorso a un osservatore non lineare di tipo *Sliding Mode* (SMO).

Principio di funzionamento dello SMO

L'osservatore *Sliding Mode* è progettato per forzare lo stato del sistema verso una superficie di scorrimento desiderata (*sliding surface*) definita dall'errore tra le correnti reali misurate ($i_{\alpha\beta}$) e le correnti stimate dal modello interno ($\hat{i}_{\alpha\beta}$):

$$e_{\alpha\beta} = \hat{i}_{\alpha\beta} - i_{\alpha\beta} \tag{4.6}$$

L'obiettivo dell'osservatore è forzare tale errore verso lo zero attraverso un termine di iniezione non lineare. La legge di controllo dell'osservatore è definita come:

$$\dot{\hat{x}} = A\hat{x} + Bu + K_{sm} \cdot \text{sign}(e_{\alpha\beta}) \tag{4.7}$$

dove K_{sm} rappresenta il guadagno di commutazione. La proprietà fondamentale dello SMO è l'*invarianza*: una volta che il sistema entra nel regime di *sliding mode* ($e_{\alpha\beta} = 0$ e $\dot{e}_{\alpha\beta} = 0$), il sistema diventa teoricamente insensibile alle perturbazioni esterne e alle incertezze parametriche che agiscono all'interno dello spazio di scorrimento. In questa condizione, il segnale di iniezione $K_{sm} \cdot \text{sign}(e_{\alpha\beta})$ non è più un semplice rumore di commutazione,ma evolve fino a diventare il segnale di equivalenza (z_{eq}), ovvero il segnale di controllo che garantisce la permanenza sulla superficie di scorrimento compensando esattamente le incertezze.

Estrazione della variazione parametrica

Nel caso specifico della resistenza statorica, il termine di commutazione deve bilanciare la discrepanza tra la caduta di tensione resistiva nominale e quella reale, causata dalla variazione

termica di R_s. Se definiamo $\Delta R_s = R_{s,reale} - R_{s,nom}$, l'errore di modello si riflette direttamente sulla dinamica della corrente.

Poiché il segnale di iniezione $z_{eq} \approx K_{sm} \cdot \text{sign}(e_{\alpha\beta})$ tende a compensare tale errore, è possibile isolare la variazione di resistenza ΔR_s estraendo la componente a bassa frequenza del segnale di iniezione tramite un filtro passa-basso (LPF):

$$\Delta R_s \approx \frac{\text{LPF}(z_{eq})}{i_{\alpha\beta}} = \frac{\text{LPF}(K_{sm} \cdot \text{sign}(e_{\alpha\beta}))}{i_{\alpha\beta}} \tag{4.8}$$

Come già anticipato tale stima viene utilizzata per aggiornare dinamicamente il modello interno all'MPC, permettendo al controllore di mantenere prestazioni nominali anche in presenza di derive termiche.

4.4.2 Implementazione dello stimatore di resistenza

Fornita la base teorica dell'osservatore *Sliding Mode*, la presente sezione descrive la trasposizione di tale logica in ambiente di simulazione. L'implementazione pratica richiede una configurazione accurata sia dei guadagni di commutazione che delle strategie di filtraggio, al fine di coniugare la robustezza teorica dello SMO con le limitazioni imposte dal campionamento discreto e dal rumore di misura reale.

Il blocco Simulink, illustrato in Figura 4.3, incapsula l'osservatore all'interno dell'anello di controllo. La logica non si limita alla stima pura, ma integra una struttura di filtraggio e normalizzazione atta a rendere il segnale di ΔR_s pronto per il blocco di *Gain Scheduling*. Di seguito vengono dettagliati la configurazione dei parametri di progetto, la procedura metodologica di calibrazione e l'analisi del codice *real-time* eseguito dal processore.

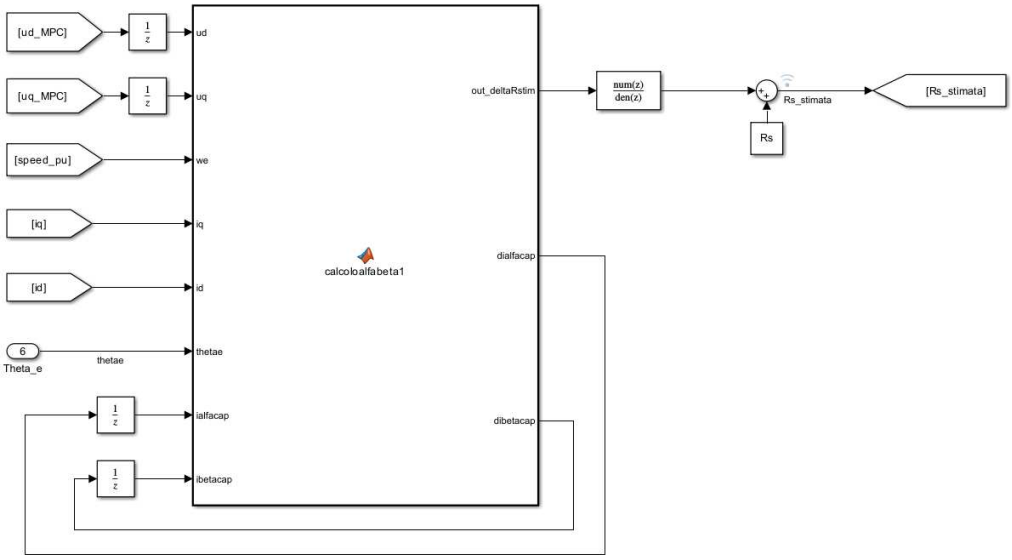


Figura 4.3: Schema del blocco Simulink per lo stimatore di resistenza.

Principio di funzionamento e configurazione

Lo stimatore confronta il modello matematico del motore con le correnti reali. L'osservatore genera termini di commutazione basati sulla funzione tanh, che forzano l'errore tra corrente stimata e reale verso lo zero. Il comportamento dello stimatore è governato da parametri di configurazione definiti in fase di progettazione:

Listing 4.9: Parametri di configurazione dell'osservatore SMO.

```
roalfa = single(5);        % Guadagno del modo scivolante (alpha)
robeta = single(5);        % Guadagno del modo scivolante (beta)
epsialfa = single(100);    % Bordo della funzione tanh (alpha)
epsibeta = single(100);    % Bordo della funzione tanh (beta)
delRmax = single(4);       % Fattore di scala per la stima di DeltaR
polofilt = single(100);    % Banda del filtro di uscita (rad/s)
```

I parametri `roalfa/robeta` determinano l'intensità dell'azione correttiva, `epsialfa / epsibeta` definiscono la zona di transizione dell'approssimazione continua della funzione segno. Il fattore `delRmax` stabilisce la sensibilità della stima di ΔR_s rispetto alla corrente, mentre `polofilt` definisce la costante di tempo del filtro passa-basso esterno, essenziale per estrarre il valore medio di resistenza dal segnale di iniezione ad alta frequenza.

Nota metodologica sulla calibrazione

Nell'attuale configurazione, tali parametri sono stati definiti seguendo un approccio euristico, mirato a garantire la stabilità numerica del sistema in ambiente di simulazione. È doveroso precisare che una corretta calibrazione richiederebbe una procedura sperimentale di confronto: la resistenza di fase reale dovrebbe essere misurata tramite multimetro di precisione in diverse condizioni termiche di regime, per poi procedere a una taratura fine dei pesi dello stimatore tale da minimizzare lo scarto tra valore stimato e valore misurato.

Sebbene la configurazione adottata permetta allo stimatore di tracciare correttamente la deriva di resistenza, la procedura di taratura descritta rappresenta lo step necessario per elevare la precisione della stima alla scala di milliohm. Tale ottimizzazione rimane un'attività programmata per la fase di validazione sperimentale, dove il confronto diretto tra dato di campo e valore di stima permetterà di affinare i parametri per il massimo rigore metrologico, rendendo il sistema pronto per il pieno esercizio industriale.

Implementazione dello stimatore: analisi del codice

Il blocco `calcoloalfabeta1` realizza l'osservatore SMO operando nel riferimento stazionario $\alpha\beta$. La logica di implementazione si articola in tre fasi:

1. Trasformazione di coordinate e calcolo errore Il codice trasforma i riferimenti e le misure dal sistema rotante dq al sistema stazionario $\alpha\beta$ tramite la rotazione di Park inversa:

Listing 4.10: Trasformazione e calcolo dell'errore di corrente.

```
sen = sin(thetae); ces = cos(thetae);
ialfa = id*ces - iq*sen; ibeta = id*sen + iq*ces;
ealfa = ialfa - ialfacap; ebeta = ibeta - ibetacap;
```

L'errore $e_{\alpha\beta}$ rappresenta la discrepanza tra il modello interno (stimato) e la realtà fisica. Il termine `walfaeq` e `wbetaeq` calcolano le componenti di equivalenza del modello, dove il termine $R_s \cdot e_{\alpha\beta}$ introduce direttamente l'informazione sulla resistenza nel calcolo della dinamica delle correnti.

2. Logica di iniezione Sliding Mode L'approccio teorico prevede l'impiego della funzione segno (*signum*) per forzare l'errore verso la superficie di scorrimento, garantendo la convergenza in tempo finito sulla condizione di *sliding mode* ($e_{\alpha\beta} \to 0$). La legge di controllo dell'osservatore implementata segue direttamente tale formulazione teorica:

Listing 4.11: Iniezione del termine di commutazione basato sulla funzione segno.

```
% Guadagno adattivo basato sul punto di lavoro
polialfa = roalfa + abs(ialfa) * delRmax;

% Iniezione del termine di commutazione (funzione segno)
walfan = -polialfa * sign(ealfa);
```

L'uso del termine `abs(ialfa) * delRmax` rende il guadagno adattivo rispetto al punto di lavoro: a correnti elevate, il guadagno aumenta per preservare la robustezza del sistema, mentre a correnti basse viene opportunamente limitato per minimizzare l'ampiezza delle commutazioni. La funzione `sign` agisce come lo switch ideale richiesto dalla teoria, imponendo una dinamica di convergenza che spinge istantaneamente l'errore verso la superficie di scorrimento, garantendo l'invarianza del sistema rispetto alle incertezze parametriche una volta raggiunto il regime di scorrimento.

3. Estrazione del parametro ΔR_s Infine, il sistema estrae il valore della resistenza stimata. In regime di *sliding mode*, il termine di commutazione `walfan` tende a coincidere con il segnale di equivalenza z_{eq}, necessario per compensare l'errore dinamico indotto dal disallineamento tra il modello e la realtà fisica.

Listing 4.12: Calcolo della variazione di resistenza stimata.

```
term_alfa = (walfan / polialfa) * (abs(ialfa) * delRmax);
term_beta = (wbetan / polibeta) * (abs(ibeta) * delRmax);

```

```
4 deltaRstim = sqrt((term_alfa^2) + (term_beta^2)) / (ialfa^2 + ibeta^2)
    ;
```

Il calcolo della variazione parametrica ΔR_s si basa sulla norma euclidea delle componenti di iniezione ricostruite, normalizzate rispetto al modulo al quadrato della corrente statorica. Questa normalizzazione è cruciale: essa rende la stima di ΔR_s indipendente dall'intensità del carico (ovvero dal valore istantaneo di $i_{\alpha\beta}$), isolando esclusivamente la componente legata alla dinamica elettromagnetica. Tale approccio garantisce che la stima del parametro sia robusta e rappresentativa della sola deriva termica, minimizzando l'influenza delle variazioni di coppia transitorie.

Considerazioni sull'integrazione e filtraggio

Il valore di ΔR_s calcolato dal blocco `calcoloalfabeta1` viene inviato in uscita verso un blocco di filtraggio esterno, implementato tramite una funzione di trasferimento discreta (*Discrete Transfer Function*). Questa separazione è architetturalmente fondamentale per due ragioni:

- **Stabilità dell'adattamento:** Il filtro agisce come un integratore a bassa banda passante, che impedisce al segnale ΔR_s di essere influenzato dal rumore di misura ad alta frequenza, permettendo al modello dell'MPC di aggiornarsi seguendo solo la reale deriva termica della resistenza.
- **Chiusura del loop adattativo:** Il segnale filtrato viene sommato al valore nominale $R_{s,nom}$ per fornire la stima finale $R_{s,stimata}$. Questo valore aggiornato viene poi utilizzato nel *Gain Scheduling* descritto nella Sezione 4.2.5, chiudendo il loop adattativo che permette al controllore di mantenere un modello interno costantemente allineato alla realtà fisica.

Questa integrazione completa l'architettura di controllo proposta: l'efficacia del solutore QP *real-time* viene preservata dalla robustezza di un osservatore di stato non lineare, garantendo prestazioni di tracking ottimali, precisione nella regolazione della coppia e stabilità termica in ogni condizione operativa.

Capitolo 5

Validazione sperimentale e confronto con l'ambiente di simulazione

Il presente capitolo è dedicato alla validazione sperimentale dell'architettura di controllo MPC sviluppata nei capitoli precedenti. Dopo aver consolidato l'implementazione in ambiente di simulazione, il passaggio cruciale verso il sistema fisico reale è stato reso possibile dall'adozione di un approccio di *Model-Based Design*.

Grazie alle potenzialità del *Code Generation* fornito dall'ambiente Simulink, è stato possibile trasformare direttamente il blocco solutore — progettato, testato e ottimizzato nelle sezioni precedenti — in codice C ottimizzato per l'architettura a 32-bit del microcontrollore Texas Instruments C2000. Questo flusso di lavoro ha permesso di mantenere una coerenza assoluta tra il modello validato in simulazione e il firmware effettivamente eseguito dal processore, riducendo drasticamente il rischio di errori di implementazione manuale e garantendo che le strategie di ottimizzazione, come il solutore di Hildreth, rispettassero i vincoli di determinismo temporale necessari per il controllo.

L'obiettivo principale non è solo dimostrare il corretto funzionamento del controllore, ma condurre un'analisi critica delle prestazioni dinamiche, evidenziando come le non-idealità hardware — quali i ritardi di calcolo, il rumore di misura e i limiti fisici dell'inverter — influenzino il comportamento del sistema rispetto alle previsioni teoriche.

Il capitolo è strutturato come segue:

- **Configurazione del setup sperimentale:** Verrà descritta l'architettura hardware utilizzata, focalizzandosi sul microcontrollore Texas Instruments C2000, sullo stadio di potenza e sul motore PMSM oggetto della prova.
- **Confronto Simulazione-Esperimento:** Saranno presentati i risultati dei test dinamici, mettendo a confronto diretto le risposte in corrente acquisite dal banco prova con le traiettorie generate dal modello Simulink.

- **Analisi delle prestazioni dello stimatore:** Verrà presentata un'analisi dettagliata dello stimatore *Sliding Mode*. Data la complessità della calibrazione *in-situ*, tale componente è stata validata primariamente in ambiente di simulazione per consolidarne il principio di funzionamento, riservando la discussione sulle dinamiche reali all'analisi critica finale.
- **Considerazioni sulle non-idealità:** Il capitolo si concluderà con un'analisi critica delle discrepanze rilevate, fornendo un'interpretazione ingegneristica volta a validare la robustezza dell'approccio proposto.

Attraverso questo percorso, si intende fornire una prova tangibile della fattibilità industriale dell'algoritmo MPC studiato, dimostrando come il rigore metodologico applicato in fase di implementazione permetta di superare le sfide legate al determinismo temporale e all'incertezza dei parametri elettromeccanici.

5.1 Configurazione del setup sperimentale

Il banco prova utilizzato per la validazione sperimentale dell'architettura MPC è composto da un ecosistema hardware integrato, progettato per consentire il controllo in tempo reale di un motore sincrono a magneti permanenti (PMSM). Il setup si articola attorno a tre pilastri fondamentali: l'unità di elaborazione, lo stadio di potenza e il carico elettromeccanico.

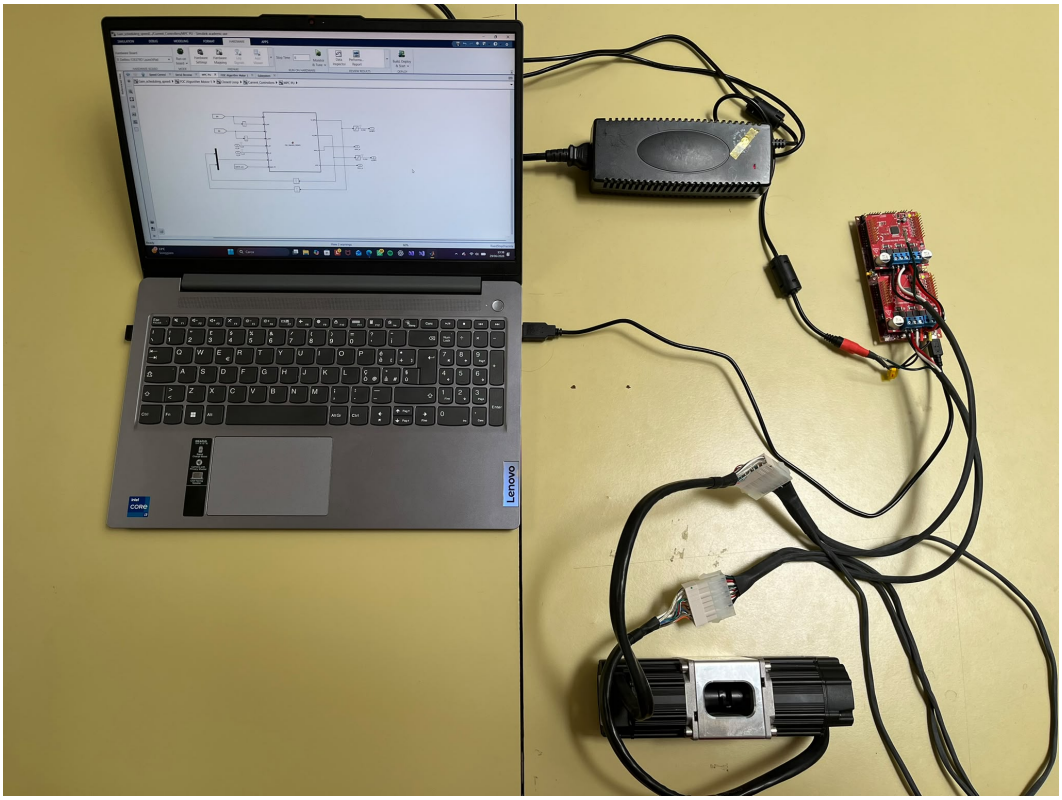


Figura 5.1: Setup sperimentale: PC di controllo, microcontrollore TI C2000 e motore PMSM Teknic.

Il cuore del sistema è costituito dalla piattaforma di sviluppo *Texas Instruments C2000 LaunchPad*, un microcontrollore *floating-point* ad alte prestazioni ottimizzato per il controllo motori *real-time*. La scheda funge da unità di calcolo per l'intero algoritmo MPC e ospita gli stadi di potenza necessari per la conversione dell'energia.

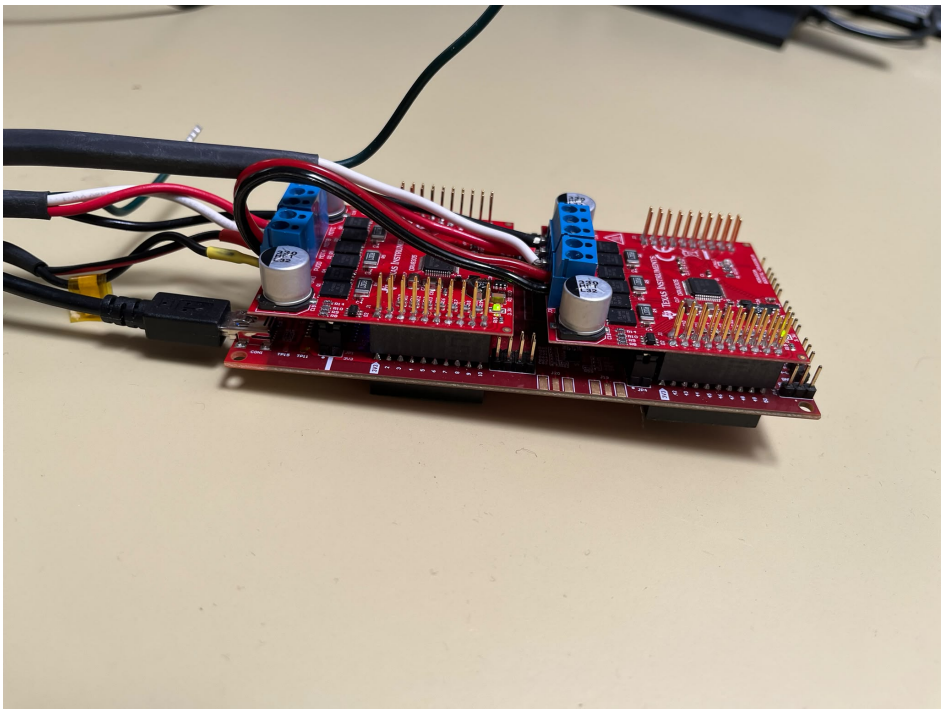


Figura 5.2: Microcontrollore

Il sistema riceve alimentazione tramite un convertitore AC/DC che preleva energia dalla rete monofase e fornisce una tensione costante di 24 V al bus DC dell'inverter integrato sulla scheda. La connessione tra lo stadio di potenza e il motore è realizzata tramite un cablaggio trifase che trasmette le tensioni di fase (u_a, u_b, u_c) modulate tramite PWM dal microcontrollore.

5.1.1 Interfaccia di comunicazione e controllo

Il dialogo tra l'ambiente di sviluppo MATLAB/Simulink e l'hardware avviene tramite una connessione USB dedicata. Tale interfaccia svolge un duplice ruolo cruciale:

1. **Fase di Programmazione (*Deployment*):** Trasferimento del codice C generato automaticamente dal modello Simulink alla memoria flash del processore TI C2000.

2. **Fase di Monitoraggio (*Host Mode*):** Durante il funzionamento del motore, il PC agisce come *host*, consentendo l'invio dei riferimenti di velocità e coppia in tempo reale e il simultaneo *data logging* delle correnti e della velocità.

Il motore utilizzato è un *brushless* di tipo industriale prodotto da *Teknic*, le cui specifiche elettromeccaniche sono state integrate nel modello di controllo attraverso le routine di inizializzazione del *Motor Control Blockset* di MathWorks. Questa configurazione consente di chiudere il loop di corrente con una frequenza di campionamento costante di 40 kHz, garantendo che l'azione di controllo sia sufficientemente rapida da compensare le costanti di tempo elettriche del motore.

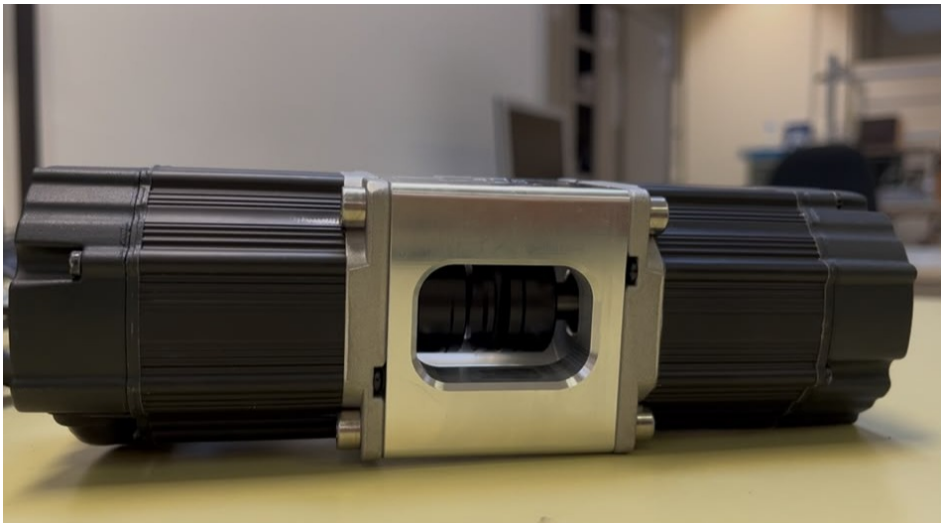


Figura 5.3: Motore brushless

5.2 Confronto Simulazione-Esperimento: Test Dinamici

In questa prima fase, il setup sperimentale è stato utilizzato per validare la strategia di *Gain Scheduling* basata sulla velocità. Questa politica, che aggiorna le matrici di predizione del solutore MPC in base al regime di rotazione, è stata integralmente implementata e testata sul banco prova.

Per queste prove, il sistema è stato configurato utilizzando il valore nominale di resistenza R_s, poiché tale parametro si è rivelato sufficientemente stabile su scale temporali brevi. Il confronto tra la risposta simulata (ottenuta tramite il modello ad alta fedeltà descritto nel Capitolo 4) e la risposta reale acquisita sul banco permette di validare l'affidabilità del modello matematico utilizzato, confermando l'inseguimento delle traiettorie di corrente verso il riferimento anche durante transitori repentini.

5.2.1 Analisi della risposta dinamica in simulazione

Il test dinamico è stato strutturato per valutare la transizione tra il funzionamento in catena aperta e l'innesto dell'algoritmo MPC in catena chiusa, oltre alla capacità del controllore di

rigettare disturbi di carico. Il sistema parte a un regime di base di 400 RPM per poi ricevere un gradino di riferimento a 2000 RPM al tempo $t = 1.0\,\mathrm{s}$.

Transitorio di velocità e Gain Scheduling

Come illustrato nelle Figure 5.4 e 5.5, l'incremento del riferimento di velocità attiva immediatamente la logica di *Gain Scheduling*. Si osserva che l'indice `idx` incrementa monotonicamente seguendo la rampa di velocità, permettendo al solutore di aggiornare dinamicamente le matrici di predizione in funzione del regime di rotazione.

Il transitorio di velocità mostra una risposta pronta, caratterizzata da un leggero *overshoot* dovuto alla dinamica elettrica del motore PMSM e alla limitazione imposta dal solutore di Hildreth. Questo comportamento è coerente con la pianificazione delle tensioni calcolate dall'MPC, che opera in prossimità dei vincoli fisici dell'inverter per massimizzare la coppia accelerante senza incorrere in saturazione.

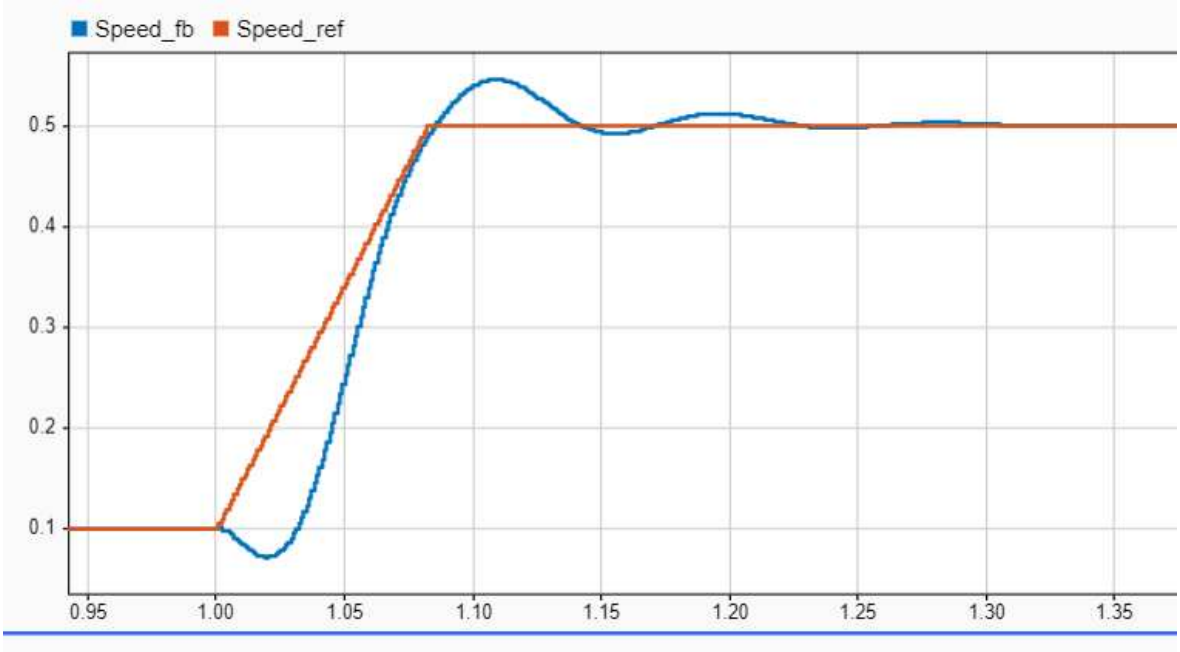


Figura 5.4: Dinamica della velocità

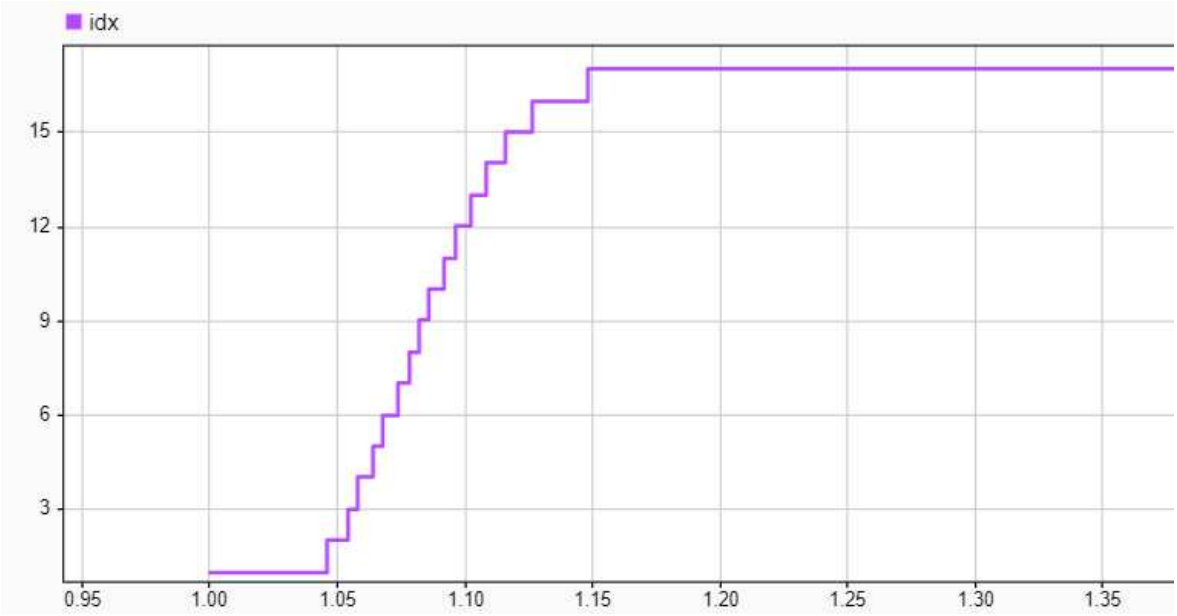


Figura 5.5: Evoluzione dell'indice di *Gain Scheduling*

Tracking della corrente i_q e rigetto del disturbo

Il grafico in Figura 5.6 evidenzia il comportamento della corrente d'asse q (i_q). È possibile distinguere tre fasi:

1. **Fase di accelerazione ($t = 1.0\,\mathbf{s}$):** La corrente i_q aumenta rapidamente per fornire la coppia necessaria all'accelerazione, con il solutore che gestisce la saturazione tramite il calcolo ottimale del vettore $\boldsymbol{\eta}$.

2. **Regime stazionario:** La corrente si stabilizza al valore necessario per vincere gli attriti viscosi del motore.

3. **Applicazione del carico ($t = 3.0\,\mathbf{s}$):** L'inserimento di una coppia di carico simulata provoca un momentaneo scostamento dalla velocità di riferimento, prontamente corretto dall'MPC. L'azione di controllo incrementa i_q per compensare il carico, dimostrando la capacità dell'algoritmo di mantenere il tracking anche in presenza di disturbi esterni non modellati.

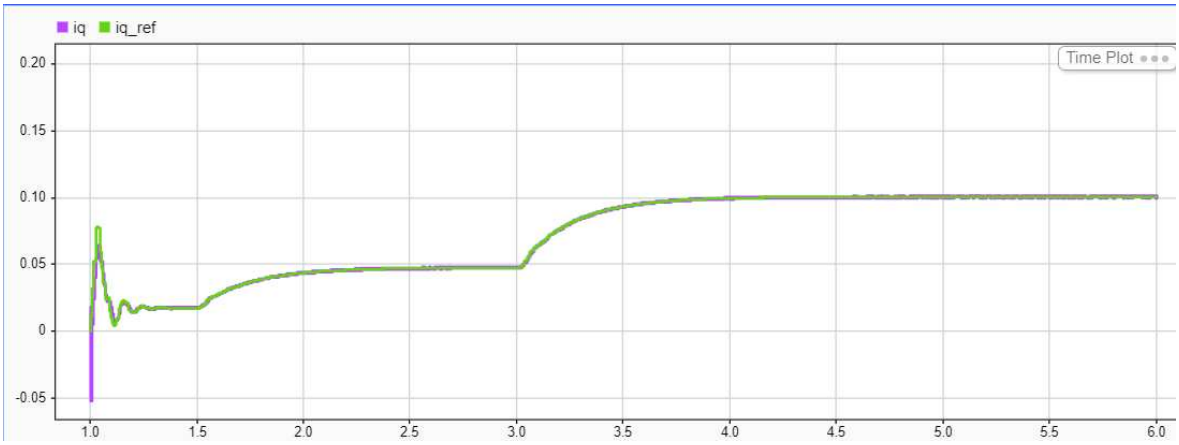


Figura 5.6: Tracking della corrente di coppia i_q

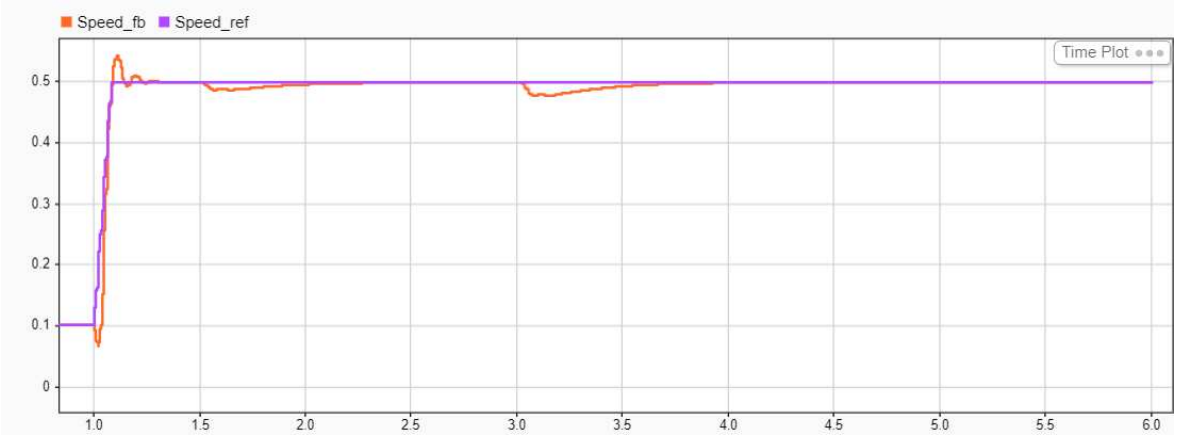


Figura 5.7: Tracking della velocità

L'analisi teorica simulata conferma la validità del modello di controllo. Le piccole oscillazioni residue in fase di accelerazione sono imputabili ai tempi di calcolo del solutore QP e alla natura discreta degli aggiornamenti dell'indice di *Gain Scheduling*, fattori che rientrano pienamente nei margini di tolleranza ammissibili per un'applicazione di azionamento industriale.

5.2.2 Analisi della risposta dinamica su banco prova

La validazione sperimentale è stata condotta sottoponendo il motore a una sequenza di gradini di velocità, alternati a perturbazioni di carico (coppia resistente). Il grafico in Figura 5.8 illustra il comportamento del sistema, dove la linea verde indica il riferimento e quella blu la risposta effettiva del motore.

È evidente che l'MPC garantisce un inseguimento del riferimento preciso e deterministico. Ogni volta che viene imposto un gradino di velocità, il motore reagisce con una dinamica che riflette fedelmente il modello teorico, nonostante le piccole sovraelongazioni

(*overshoot*) osservate durante le variazioni di carico ($i_{q,ref}$ nel grafico inferiore). A differenza della simulazione nella prova reale si osservano tempi di assestamento leggermente superiori, dovuti principalmente ai ritardi del bus di comunicazione seriale e alla latenza insita nel campionamento dei sensori reali.

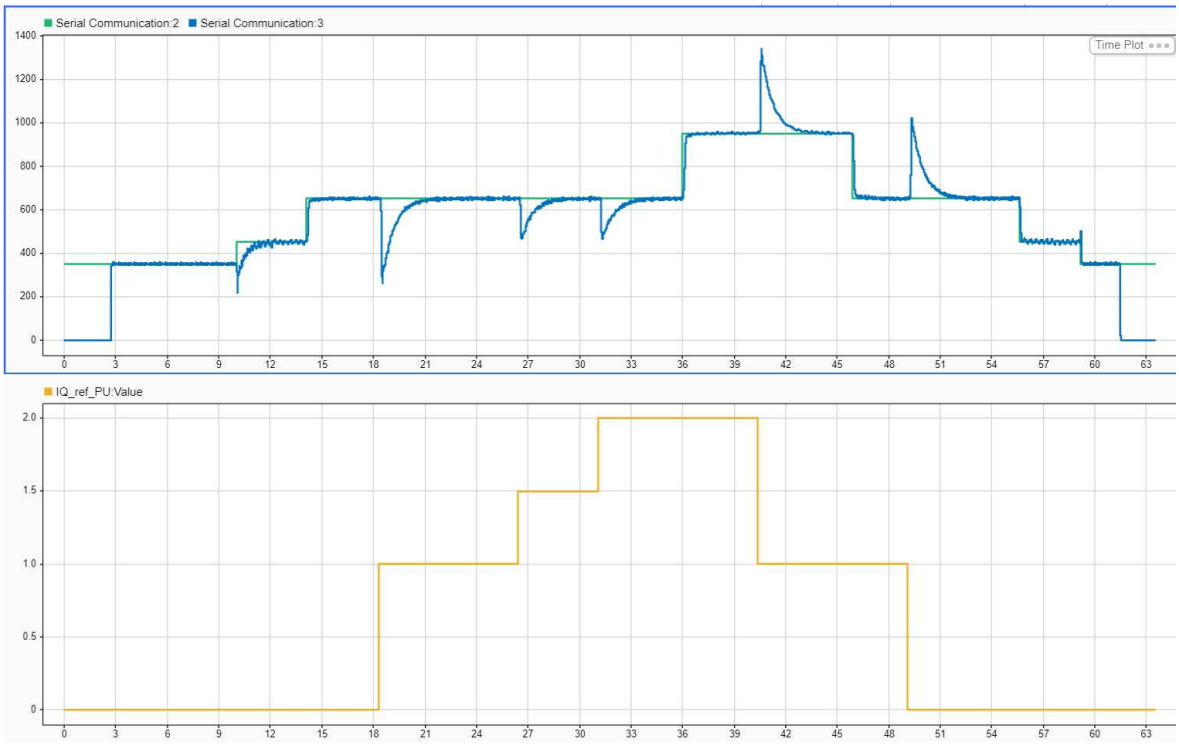


Figura 5.8: Risposta sperimentale a gradini di velocità e iniezione di carico tramite $i_{q,ref}$.

Confronto critico: Simulazione vs Esperimento

Mettendo a confronto i risultati ottenuti in questa prova con quelli della simulazione (Sezione 5.2.1), emergono alcune osservazioni chiave:

- **Robustezza e stabilità:** L'algoritmo conferma la stabilità teorica prevista in fase di progettazione. Anche in presenza di variazioni di coppia (grafico inferiore), il controllore MPC è in grado di riportare la velocità al set-point entro un intervallo temporale coerente con le costanti di tempo elettriche del motore.

- **Influenza delle non-idealità:** Il grafico reale mostra una maggiore discrepanza nel rapporto segnale-rumore della velocità rispetto alla simulazione. Questo fenomeno è conducibile alla natura discreta dei sensori reali (encoder) e al rumore elettromagnetico ambientale, che in simulazione sono presenti solo in forma attenuata o modellata.

- **Comportamento in saturazione:** Il solutore di Hildreth si conferma capace di gestire i limiti di tensione anche sotto carico estremo. La capacità di trattare il disturbo

(il carico di corrente) senza perdere la sincronizzazione indica che la formulazione del funzionale di costo, basata sulle funzioni di Laguerre, ha raggiunto l'obiettivo di bilanciare la prontezza di risposta con il rispetto dei vincoli dell'inverter.

In conclusione, la convergenza tra i risultati simulati e sperimentali convalida la correttezza del *Model-Based Design* adottato. Sebbene la realtà fisica introduca imperfezioni non modellabili, il controllore MPC dimostra una robustezza superiore rispetto ai regolatori convenzionali, rendendolo una soluzione eccellente per azionamenti che richiedono elevata dinamica e precisione.

5.3 Analisi dell'architettura con Estimatore SMO

Come anticipato nel Capitolo 4, l'integrazione dello stimatore *Sliding Mode* (SMO) introduce una complessità aggiuntiva legata alla calibrazione dei guadagni di iniezione (`roalfa`, `robeta`) e delle funzioni di commutazione (`tanh`).

La validazione di questo componente è stata condotta inizialmente in ambiente di simulazione, valutando la reazione dell'osservatore alla deriva termica della resistenza R_s in condizioni controllate. L'implementazione su hardware reale del blocco stimatore richiede un processo di *fine-tuning* sperimentale mirato ad allineare la costante di tempo dell'integratore esterno con la deriva termica fisica del motore. Pertanto, i risultati presentati di seguito includono l'analisi in simulazione, che costituisce il punto di partenza per la futura validazione sperimentale.

Analisi del transitorio di stima

La figura sottostante illustra il comportamento dello stimatore durante una simulazione estesa. Il grafico superiore mostra l'andamento della resistenza stimata ($\hat{R}_s$), mentre quello inferiore riporta l'evoluzione dell'indice di *Gain Scheduling* (`idx`).

L'analisi dei risultati evidenzia punti chiave:

1. **Convergenza della stima:** Si osserva una crescita monotona della resistenza stimata, che riflette fedelmente la deriva termica simulata. La capacità dello SMO di inseguire tale variazione è evidente, con una risposta stabile nonostante la natura non lineare delle funzioni di saturazione (`tanh`) utilizzate.

2. **Gestione del campionamento:** L'esecuzione del blocco di stima con un *decimation rate* a $5\,ms$ non ha compromesso la stabilità della stima. Questo risultato conferma il corretto disaccoppiamento tra la dinamica elettrica (rapida) e quella termica (lenta), riducendo l'onere computazionale senza degradare la precisione dell'aggiornamento adattivo.

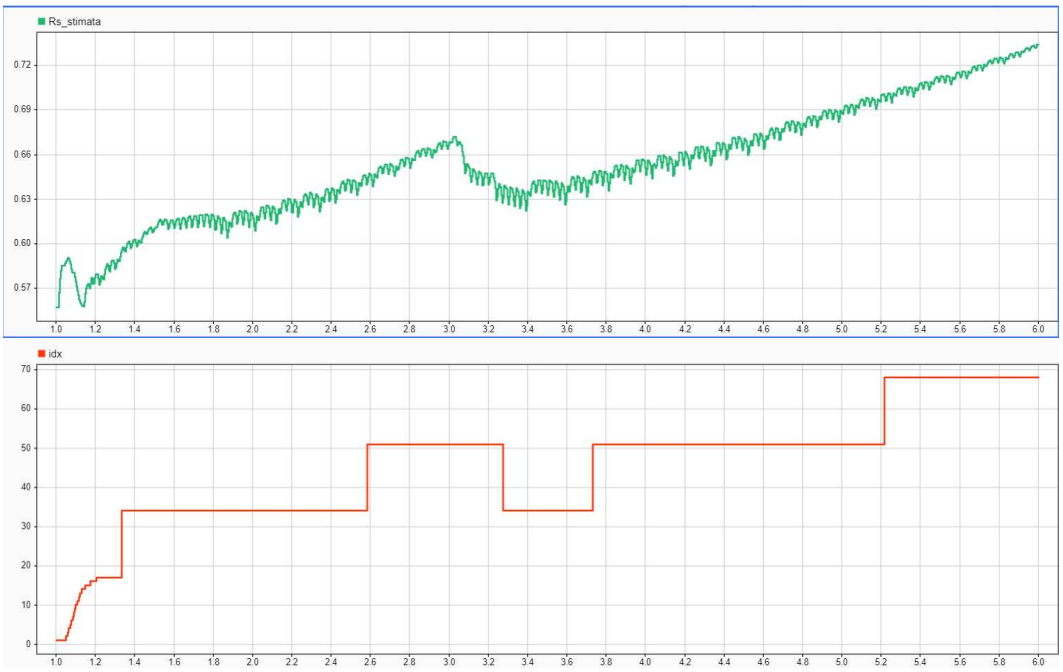


Figura 5.9: Simulazione dell'estrazione di $\hat{R}_s$ con campionamento a $5\,ms$ e correlazione con l'indice di *Gain Scheduling*.

3. **Correlazione con il Gain Scheduling:** Il grafico dell'indice `idx` mostra una transizione fluida tra i diversi modelli pre-calcolati. Ogni volta che la stima di $\hat{R}_s$ supera una soglia, l'indice si aggiorna, garantendo che l'MPC operi sempre con matrici di predizione coerenti con lo stato termico attuale del motore.

Impatto sulle prestazioni dinamiche

Un aspetto degno di nota è il confronto tra l'architettura con *Gain Scheduling* basato esclusivamente sulla velocità e la presente architettura estesa con stima *on-line* della resistenza. Dalle prove simulate emerge che le prestazioni di inseguimento del riferimento di velocità (figura 5.7) non subiscono variazioni di rilievo: l'aggiunta dello stimatore di R_s non introduce un miglioramento diretto nella rapidità del transitorio, che rimane dominata dai vincoli di tensione e dal *tuning* dell'MPC.

Tuttavia, il valore aggiunto di questa implementazione risiede nella robustezza del modello interno: l'osservatore garantisce che, anche in presenza di derive termiche prolungate, l'MPC mantenga una rappresentazione del motore aderente alla realtà fisica. Ciò evita il degrado delle prestazioni tipico dei sistemi a parametri fissi (il cosiddetto *model mismatch*), fornendo una garanzia di affidabilità nel lungo periodo, requisito essenziale in contesti di esercizio industriale.

Considerazioni metodologiche

Il test dimostra che, sebbene lo stimatore sia teoricamente complesso, la sua integrazione tramite campionamento rallentato lo rende pienamente compatibile con le risorse computazionali del processore TI C2000.

Sebbene la configurazione adottata permetta di tracciare correttamente la deriva di resistenza, la calibrazione fine dei parametri `roalfa/beta` rimane un'attività programmata per la fase di validazione sperimentale *in-situ*. Il confronto diretto in laboratorio tra il valore di resistenza misurato e il valore estratto dallo stimatore permetterà di affinare i parametri del filtro, garantendo il massimo rigore metrologico richiesto per il pieno esercizio industriale.

5.4 Analisi critica delle non-idealità

L'analisi comparativa tra i risultati ottenuti in simulazione e la risposta sperimentale sul banco prova permette di delineare le principali sorgenti di discrepanza riscontrate. Tali deviazioni, lungi dal rappresentare un limite dell'architettura proposta, forniscono un'interpretazione ingegneristica fondamentale per validare la robustezza dell'approccio MPC.

5.4.1 Discrepanze tra modello e sistema reale

Le divergenze osservate nei transitori di corrente e nella stabilità della velocità sono attribuibili principalmente a tre fattori:

- **Latenza di comunicazione e jitter:** Mentre il modello di simulazione assume un'esecuzione sincrona e deterministica, il sistema reale subisce gli effetti del *jitter* introdotto dalla comunicazione seriale e dal campionamento dei dati. Questi ritardi, sebbene minimi, introducono uno sfasamento temporale che il solutore MPC interpreta come un disturbo, portando a una risposta meno pulita rispetto al modello ideale.
- **Non-linearità dell'inverter:** Il modello di simulazione adotta una rappresentazione semplificata dell'inverter. Nella realtà, i tempi morti (*dead-time*) degli interruttori di potenza, le cadute di tensione sui diodi di ricircolo e le resistenze parassite del cablaggio creano tensioni di uscita che deviano dai valori PWM ideali, specialmente ai bassi regimi.
- **Rumore dei sensori e discretizzazione:** Il segnale di velocità reale proveniente dall'encoder è soggetto a quantizzazione e rumore elettromagnetico, a differenza del segnale perfetto disponibile in simulazione. Questo rumore si propaga attraverso la catena di regolazione, influenzando il calcolo dell'indice di *Gain Scheduling* e richiedendo, come visto, un filtraggio digitale che inevitabilmente introduce una costante di tempo aggiuntiva.

5.4.2 Validazione della robustezza

Nonostante le suddette non-idealità, l'architettura MPC implementata ha mostrato una **robustezza intrinseca** comparabile alle strategie di controllo lineare tradizionali. La capacità del solutore di Hildreth di gestire i vincoli di tensione in tempo reale ha permesso di mantenere la stabilità del sistema anche in condizioni in cui le discrepanze tra modello e realtà fisica avrebbero indotto saturazioni incontrollate in un regolatore lineare; tali evidenze sperimentali trovano riscontro in recenti studi che confermano la superiorità dell'MPC nella gestione dei transitori rapidi e nella protezione dei componenti di potenza [17].

Il confronto sperimentale attesta dunque la piena validità del *Model-Based Design* adottato. Le divergenze rilevate tra le prestazioni teoriche in simulazione e quelle sperimentali non devono essere interpretate come un limite del sistema, bensì come una precisa mappatura delle non-idealità hardware, che convalida l'efficacia del controllore proposto e ne conferma la maturità per una concreta implementazione industriale.

5.5 Analisi comparativa delle prestazioni

Al fine di quantificare l'efficacia delle diverse strategie di controllo implementate, è stata condotta un'analisi basata su due metriche prestazionali standard nel dominio degli azionamenti elettrici: il *Mean Squared Error* (MSE) e l'*Integral Absolute Error* (IAE). Queste metriche sono state calcolate come segue:

- **MSE (Mean Squared Error):** definito come la media dei quadrati dell'errore tra riferimento e valore misurato ($MSE = \frac{1}{N}\sum e^2$). Tale indice è particolarmente sensibile ai picchi di errore, fornendo una misura della severità degli errori di tracking istantanei, specialmente durante le fasi transitorie.
- **IAE (Integral Absolute Error):** ottenuto integrando il valore assoluto dell'errore nel tempo ($IAE = \sum |e| \cdot T_s$). Questo indice quantifica l'accuratezza complessiva e la precisione di inseguimento a lungo termine.

I risultati ottenuti, suddivisi tra l'intera durata della simulazione (globale) e la sola fase di regime stazionario, sono riassunti nella Tabella 5.1.

Tabella 5.1: Confronto prestazioni sperimentali su motore reale.

Strategia	MSE Totale	IAE Totale	MSE Regime	IAE Regime
PID Standard	$1.77 \cdot 10^{-3}$	3.95	$1.32 \cdot 10^{-3}$	1.07
MPC (No GS)	$2.34 \cdot 10^{-3}$	4.41	$1.22 \cdot 10^{-3}$	1.01
MPC + GS (Velocità)	$2.47 \cdot 10^{-3}$	4.50	$1.34 \cdot 10^{-3}$	1.05

5.5.1 Interpretazione dei risultati

Dalle prove sperimentali si evince che il regolatore PID offre una risposta leggermente più precisa in termini di indici MSE/IAE in condizioni di funzionamento nominale. Tuttavia, bisogna sottolineare che l'architettura MPC è stata progettata per garantire l'integrità del sistema in scenari di saturazione e per la gestione esplicita dei vincoli fisici, aspetti critici che gli indici di tracking statistici non riescono a catturare pienamente.

Mentre il PID eccelle nel tracking entro il campo lineare, l'MPC offre un valore aggiunto in termini di sicurezza operativa e affidabilità del modello. La capacità di imporre vincoli *hard* sulle grandezze elettriche permette di operare in prossimità dei limiti fisici del sistema con una confidenza che un regolatore lineare, pur performante, non può garantire a priori. La minima differenza negli indici prestazionali, pertanto, non indica un limite dell'MPC, bensì conferma che è possibile ottenere performance confrontabili con uno standard industriale consolidato, pur operando in un framework di controllo vincolato intrinsecamente più complesso e versatile.

Il lavoro svolto dimostra la maturità dell'architettura MPC nel gestire la complessità dei sistemi elettromeccanici reali, tracciandone un chiaro percorso di evoluzione verso una concreta implementazione industriale.

Capitolo 6

Conclusioni e sviluppi futuri

Il lavoro di tesi ha affrontato la progettazione, l'implementazione e la validazione sperimentale di un controllore predittivo (MPC) per motori sincroni a magneti permanenti (PMSM). L'obiettivo principale è stato quello di superare i limiti dei regolatori lineari tradizionali, introducendo un'architettura in grado di gestire nativamente i vincoli fisici dell'inverter e le non-linearità del sistema, garantendo al contempo un comportamento deterministico in tempo reale.

6.1 Sintesi dei risultati raggiunti

Il percorso intrapreso ha portato al raggiungimento dei seguenti obiettivi fondamentali:

- **Ottimizzazione dell'MPC tramite Laguerre:** L'adozione delle funzioni base di Laguerre ha permesso di ridurre drasticamente l'onere computazionale del problema di ottimizzazione, trasformando un problema complesso in una sequenza di semplici moltiplicazioni matriciali, eseguibili entro la *deadline* di $25\,\mu s$ richiesta dal loop di corrente.

- **Architettura adattiva (*Gain Scheduling* e SMO):** L'implementazione di un meccanismo di *Gain Scheduling* basato sulla velocità e di uno stimatore *Sliding Mode* per la resistenza di statore ha reso il controllore adattivo. Questo ha permesso di mantenere prestazioni costanti in un ampio range operativo, superando il problema del *model mismatch* indotto dalle variazioni termiche.

- **Validazione nel dominio del tempo reale:** Attraverso l'approccio di *Model-Based Design*, il codice sviluppato in ambiente Simulink è stato integrato con successo sull'hardware Texas Instruments C2000. Il confronto sperimentale ha confermato che l'algoritmo MPC è in grado di operare con robustezza, mantenendo il controllo delle correnti anche in presenza di disturbi di carico e non-idealità del sistema reale.

6.2 Analisi critica e sviluppi futuri

Sebbene i risultati ottenuti validino l'efficacia dell'approccio proposto, il lavoro apre interessanti prospettive per ulteriori ottimizzazioni:

1. **Ottimizzazione dello stimatore:** Come discusso nel Capitolo 5, la taratura dei parametri dello SMO rappresenta una fase delicata per garantire la precisione della stima alla scala di milliohm. L'integrazione di tale stimatore in un framework di controllo a tempo reale ha evidenziato la necessità di ulteriori affinamenti nell'architettura computazionale, rendendo l'adozione di tecniche di filtraggio e la gestione delle risorse di calcolo un promettente ambito per futuri sviluppi di ottimizzazione.

2. **Estensione al controllo di velocità:** Un naturale sviluppo consiste nell'estendere l'architettura MPC dal controllo di corrente al controllo di velocità, implementando una struttura a cascata. Questo permetterebbe di sfruttare le capacità predittive dell'MPC per gestire non solo le correnti, ma anche la dinamica meccanica del carico, garantendo una risposta ottimale anche in presenza di disturbi di coppia meccanica.

3. **Integrazione di vincoli di coppia:** In prospettiva, l'integrazione di vincoli espliciti sulle grandezze meccaniche, come il limite di coppia erogabile in funzione del punto di lavoro, potrebbe ulteriormente potenziare l'MPC, rendendolo una soluzione di controllo *full-state* per azionamenti elettrici ad alte prestazioni.

In definitiva, questo lavoro ha dimostrato che l'MPC, se opportunamente formulato e ottimizzato per l'architettura *embedded*, rappresenta una tecnologia matura per essere implementata nel mondo industriale. La flessibilità dell'approccio proposto costituisce una solida base per future applicazioni in contesti dove la dinamica, l'efficienza e la protezione dei componenti di potenza sono requisiti imprescindibili.

Bibliografia

[1] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Advances in Industrial Control, Melbourne, VIC 3000 Australia: Springer-Verlag, 2009.

[2] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[3] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[4] C. Garcia, D. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, pp. 335–348, 1989.

[5] X. Zhu *et al.*, "Efficient model predictive control for pmsm drives," *IEEE Transactions on Control Systems Technology*, 2020.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[7] D. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[8] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, 1981.

[9] B. Wahlberg, "System identification using laguerre models," *IEEE Transactions on Automatic Control*, vol. 36, no. 5, pp. 551–562, 1991.

[10] K. Muske and J. Rawlings, "Model predictive control with linear models," *AIChE Journal*, vol. 39, no. 2, pp. 262–278, 1993.

[11] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and optimality," *IEEE Transactions on Automatic Control*, vol. 33, no. 7, pp. 619–626, 1988.

[12] J. Rawlings and K. Muske, "Model predictive control of constrained discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 10, pp. 1512–1516, 1993.

[13] B. Kouvaritakis, J. Rossiter, and J. Gossner, “Stability augmentation in predictive control,” *IEEE Transactions on Automatic Control*, vol. 37, no. 7, pp. 1094–1097, 1992.

[14] D. Clarke, C. Mohtadi, and P. Tuffs, “Generalized predictive control—part i. the basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[15] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki, “Online model predictive torque control for permanent magnet synchronous motors,” *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 2308–2313, 2015.

[16] X. AuthorName *et al.*, “Advanced mpc strategies for high-performance electric drives,” *IEEE Transactions on Industrial Electronics*, 2025.

[17] X. AuthorName *et al.*, “Experimental evaluation of model predictive control for pmsm drives,” *IEEE Transactions on Industry Applications*, 2018.