

TRABAJO ESPECIAL DE GRADO

DESARROLLO DE UN ALGORITMO DE CONTROL PARA UN ACCIONAMIENTO AC BRUSHLESS CON SONDAS A EFECTO HALL DIGITALES

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Martínez V., Manuel A.
para optar al título de
Ingeniero Electricista

Caracas, 2008

TRABAJO ESPECIAL DE GRADO

DESARROLLO DE UN ALGORITMO DE CONTROL PARA UN ACCIONAMIENTO AC BRUSHLESS CON SONDAS A EFECTO HALL DIGITALES

Prof. Guía: Ing. Julio Molina
Tutor Industrial: Prof. Ing. Fabio Giulii Capponi
Ing. Giulio De Donato

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Martínez V., Manuel A.
para optar al título de
Ingeniero Electricista

Caracas, 2008

Martínez V., Manuel A.

**DESARROLLO DE UN ALGORITMO DE CONTROL PARA UN
ACCIONAMIENTO AC BRUSHLESS CON SONDAS A EFECTO HALL
DIGITALES**

Tutor Académico o Prof. Guía: Ing. Julio Molina. **Tutor Industrial:** Prof. Ing. Fabio Giulii Capponi e Ing. Giulio De Donato. **Tesis.** Roma. **Università di Roma – “La Sapienza”.** Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. **Ingeniero Electricista. Opción: Potencia. 2008. 172 h.**

Palabras Claves: Accionamiento Eléctrico, Electrónica Industrial de Potencia y Maquinas Eléctricas.

Resumen. Se plantea el estudio teórico y la implementación del software de un observador de velocidad y posición angular dentro de un DSP (Digital Signal Processor) para un accionamiento del tipo AC Brushless que utiliza 2 sondas a efecto Hall digitales para la medición de la posición angular. Se presenta un resumen sobre los accionamientos eléctricos donde se describe la maquina eléctrica sincrónica con excitación a través de imanes permanentes (**SPM** - *surface permanent magnet* y **IPM** - *interior permanent magnet*), con particular énfasis en los accionamientos AC Brushless y DC Brushless. Se describe el procedimiento utilizado para la estima de la posición y velocidad mediante un Encoder compuesto por 2 sondas a efecto Hall de baja resolución colocadas a 90° eléctricos entre ellas, para esto se utiliza un Vector Tracking Observer VTO (Observador a seguimiento vectorial). Se emplea el hardware y software del Digital Signal Processor (DSP) para implementar el observador, en la descripción del hardware se hace referencia a la estructura, características técnicas y periféricas utilizadas; en la descripción del software se presenta el algoritmo utilizado para la programación del observador. Por último, se presentan las simulaciones hechas en Matlab y los resultados experimentales obtenidos de la implementación del observador mediante el DSP. El argumento desarrollado en esta tesis es de gran interés científico e industrial a nivel internacional en el sector de los accionamiento eléctricos. Grandes esfuerzos se realizan por parte de los principales grupos de investigación internacionales, tanto a nivel universitario como industrial, para encontrar alternativas válidas para la utilización de sensores de posición angular a elevada resolución para accionamientos AC Brushless.



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria

Dipartimento di Ingegneria Elettrica

Tesi di Laurea Specialistica

"Sviluppo di algoritmi di controllo per un azionamento AC Brushless con sonde ad effetto Hall digitali"

Candidato

Manuel Alejandro Martinez Vaamonde

Relatore

Prof. Ing. Fabio Giulii Capponi

Correlatore

Ing. Giulio De Donato

ANNO ACCADEMICO 2006-2007

Ringraziamenti

Desde el primer momento que comencé mis estudios de Ingeniería Eléctrica sabía que no sería fácil llegar al final, pero ha sido maravilloso el desarrollo de toda la carrera; además quiero agradecer a mi “Dios todopoderoso”, a la “Virgen del Valle” y a todos los santos por darme la fuerza, voluntad y salud necesaria para cumplir con esta misión que hoy completo.

Gracias a Dios que cuento con una madre, un padre y una hermana que lo han dado todo por mí, que siempre han creído en mi sueño de ser Ingeniero Electricista; mi madre Angela Vaamonde ha sido realmente espectacular, no solo por estar siempre a mi lado sino porque desde que nací me ha dado todo lo que necesitaba para ser un hombre correcto e independiente, “Madre no tengo palabras con que describir lo feliz que me siento de ser tú hijo, este título te lo dedico con todo mi amor”. Mi padre Juan Martinez ha sido más que un padre, realmente un gran amigo para mí porque desde pequeño me enseñaste que estudiar y hacer deportes me ayudaría a ser un hombre de bien, “Padre gracias por todos los consejos que siempre me has dado, este título también es tuyo”. Mi hermanita bella María de los Angeles ha sido el regalo más grande que Dios y mis padres me han podido dar, siendo menor que yo siempre me has cuidado y protegido, “Mi gorda bella gracias por todo el amor y esa maravillosa sonrisa que me has dado desde que naciste”.

Mi familia también ha sido un gran apoyo para mí porque desde mis abuelos hasta mis primos más pequeños siempre han estado conmigo, tanto en los mejores momentos de mi vida como los que por un motivo u otro no han sido tan buenos; de verdad que le doy gracias a Dios por cada uno de ustedes. A mi novia Maigli Marleth, por haber tenido la paciencia y la voluntad necesaria para soportar estar tan lejos y seguir creyendo en este amor que nos une, “Gracias nena por tu apoyo incondicional y todo el amor que me has dado desde que nos hicimos novios”.

Ringrazio alla “Universidad Central de Venezuela” e la “Università di Roma – La Sapienza” per darmi l'opportunità di partecipare in questa convenzione e poter ottenere la doppia titolazione in Ingegneria Elettrica. Per fortuna ho conosciuto al mio relatore, Prof. Ing. Fabio Giulii Capponi, insieme a lui ho potuto lavorare nell'area degli azionamenti elettrici, essendo questa area del mio interesse; grazie a lui ho potuto conoscere e lavorare insieme al mio correlatore, Ing. Giulio De Donato, senza l'aiuto di loro due non sarebbe stato possibile finire questa tesi. Inoltre, devo ringraziare a tutto il team del laboratorio di macchine (Paolo, Alessandro, Antonio, Alessio, Eugenio, Adriano) perché l'ambiente di lavoro è stato veramente eccezionale.

Para finalizar, quiero compartir este título con todos mis compañeros que se encuentran igual que yo realizando el convenio de doble titulación, a la primera generación (Johnny, Ambar, Irene, Hector y Freddy), la segunda generación (Luis, Joao, Andrea, Raymond, William y Tato) y la tercera generación (Cynthia, Jesus M., Yiorio Harmany, Viva la pepa, Richard, Jesus R., Carmen, Mercedes) porque con ellos he pasado los mejores momentos de mi estadía en Roma, “Gracias

RINGRAZIAMENTI

muchachos por darme la fuerza y el apoyo para seguir adelante". En particular agradezco a mi colega, Ing. Freddy Da Silva, con quien compartí la mayor parte de mi carrera y hoy estamos celebrando que logramos graduarnos.

Indice

<i>Ringraziamenti</i>	<i>ii</i>
<i>Introduzione</i>	<i>1</i>
I.1. Profilo del progetto	1
I.2. Descrizioni dei capitoli	2
<i>Capitolo 1: Riassunto sugli azionamenti elettrici</i>	<i>4</i>
1.1. Introduzione	4
1.2. Motore Sincrono a Magneti Permanenti	4
1.3. Gli Azionamenti Elettrici	7
1.3.1. Caratteristiche generali di un Azionamento Elettrico	7
1.3.2. Struttura di un Azionamento Elettrico	8
1.3.2.1. Motore	8
1.3.2.2. Convertitore	8
1.3.2.3. Trasmissione	9
1.3.2.4. Controllore	9
1.3.2.5. Sistema di controreazione	9
1.4. Trasduttori de posizione angolare e velocità	9
1.4.1. Risoluzione, Accuratezza assoluta ed Accuratezza differenziale	9
1.4.2. Trasduttori di posizione angolare	12
1.4.2.1. Encoder incremental	12
1.4.2.2. Encoder assoluto	15
1.4.2.3. Resolver	16
1.4.2.4. Sonde ad effetto hall	18
1.4.2.4.1. Sonda ad effetto Hall a misura diretta	19
1.4.2.4.2. Sonda ad effetto Hall a compensazione di campo	20

1.4.2.4.3. Encoder composto da sonde ad effetto Hall	21
1.5. Azionamento con Motori Brushless	22
1.5.1. Generalità	22
1.5.2. Modello della macchina sincrona	23
1.5.2.1. Struttura fisica ed equazioni descrittive	23
1.5.2.2. Trasformazioni delle variabili	25
1.5.2.3. Bilancio energetico nel dominio di Park	27
1.5.3. Modello del Motore Brushless	28
1.5.4. Azionamenti con motori "AC Brushless"	30
1.5.5. Azionamenti con motori "DC Brushless"	34
1.6. Conclusioni	39
 Capitolo 2: Procedimenti per la stima di velocità e posizione	 40
2.1. Introduzione	40
2.2. Elaborazione dei segnali attraverso 2 sonde ad effetto Hall	40
2.3. Vector-Tracking Observer VTO	46
2.3.1. Struttura dell'osservatore	46
2.3.2. Analisi del punto di lavoro (linearizzazione dell'osservatore)	48
2.3.3. Sintesi dell'osservatore in tempo discreto	48
2.4. Applicazione dell'osservatore per un sistema di misura di posizione e velocità attraverso sonde ad effetto Hall	55
2.4.1. Problemi di quantizzazione	55
2.4.2. Gain Scheduling	61
2.4.3. Disaccoppiamento delle armoniche di quantizzazione	66
2.5. Conclusioni	76

Capitolo 3: Hardware e software del Digital Signal Processor (DSP) per l'implementazione dell'osservatore	77
3.1. Introduzione	77
3.2. Digital Signal Processors DSP (eZdsp™ TMS320F2812)	77
3.2.1. Caratteristiche tecniche del DSP	77
3.2.2. Struttura del DSP	78
3.2.2.1. Scheda	80
3.2.2.1.1. Alimentazione	80
3.2.2.2. Memoria	80
3.2.2.2.1. Mappa della memoria	81
3.3. Struttura e routine del software per l'implementazione dell'osservatore	82
3.3.1. Periferiche e librerie del DSP (eZdsp™ TMS320F2812) utilizzate per l'implementazione dell'osservatore	83
3.3.1.1. Event Manager (EV)	83
3.3.1.2. Analog-Digital Converter (ADC)	85
3.3.1.3. Serial Peripheral Interface (SPI)	87
3.3.1.4. Fixed Point Math Library (Qmath Library)	90
3.3.1.5. Virtual Floating Point Engine (IQmath Library)	91
3.3.2. Routine di inizio (void main(void))	94
3.3.3. Routine di inizializzazione del EVA (void init_eva(void))	97
3.3.4. Routine di inizializzazione dei registri della SPI FIFO (void spi_fifo_init(void))	98
3.3.5. Routine degli indirizzi delle variabili inviate al DAC "AD7568" (void AD7568_init(void))	99
3.3.6. Routine per l'invio dei dati al DAC "AD7568" (void AD7568_write (int16 buf[8]))	99
3.3.7. Routine principale di controllo del motore (interrupt void motorcontrol(void))	100
3.3.7.1. Pulse-Width Modulation PWM (Modulazione ad ampiezza d'impulsi)	100

3.3.7.2. Trasformazione di coordinate per le Correnti di fase (ABC) → Corrente nel sistema di riferimento fisso ($\alpha\beta$)	106
→ Corrente nel sistema di riferimento arbitrario (qd0)	
3.3.7.3. VECTOR TRACKING OBSERVER VTO (Osservatore ad inseguimento vettoriale)	109
3.3.7.4. Digital-Analog Converter DAC	114
3.4. Conclusioni	115
 Capitolo 4: Simulazioni e prove sperimentali sull'osservatore	 116
4.1. Introduzione	116
4.2. Impostazione dell'osservatore attraverso Matlab	116
4.3. Risultati delle simulazioni attraverso Matlab	117
4.3.1. Posizione Angolare Stimata (Velocità Angolare 10 rad/s)	117
4.3.2. Posizione Angolare Stimata (Velocità Angolare 200 rad/s)	120
4.3.3. Posizione Angolare Stimata (Velocità Angolare 400 rad/s)	123
4.4. Impostazione dell'osservatore sul DSP eZdsp TMS320F2812	126
4.5. Risultati sperimentali delle prove all'osservatore attraverso il DSP eZdsp TMS320F2812	128
4.5.1. Posizione Angolare Stimata (Velocità Angolare 10 rad/s)	128
4.5.2. Posizione Angolare Stimata (Velocità Angolare 200 rad/s)	131
4.5.3. Posizione Angolare Stimata (Velocità Angolare 400 rad/s)	134
4.5.4. Pulse-Width Modulation PWM	137
4.5.5. Corrente nel sistema di riferimento arbitrario (qd0)	139
4.6. Conclusioni	141
 Conclusioni generali	 142

<i>Appendice</i>	<i>143</i>
A.1. DSP281x_Constant.h	143
A.2. DSP281x_Device.h	144
A.3. DSP281x_Examples.h	146
A.4. DSP281x_AD7568.h	148
A.5. DSP281x_Osservatore.c	148
 <i>Bibliografia</i>	 <i>164</i>

Introduzione

I.1. Profilo del progetto

“Gli azionamenti elettrici” sono una vasta gamma di sistemi composti da attuatori elettrici (i motori) utilizzati per generare il movimento, dai rispettivi moduli di controllo, nonché degli organi di trasmissione e trasformazione del moto. Questi azionamenti sono utilizzati nei settori applicativi che comprendono, tra l'altro, elettrodomestici, trazione ferroviaria, impiantistica industriale e civile (pompe, ascensori, ventilatori, ecc.), robot industriali, macchine a controllo numerico, strumentazione elettronica, periferiche di calcolatori (testine di stampanti, unità dischi, scanner, ecc.), accessori automobilistici (vetri elettrici, tergicristalli, pompe della benzina), industria aerospaziale e molti altri. Ciascuna di queste applicazioni ha requisiti differenti in termini di potenza installata e caratteristiche di comando.

Talvolta i motori funzionano prevalentemente a velocità e carico costanti o prefissati o al massimo selezionati tra un piccolo insieme di valori (si pensi, ad esempio, alle pompe, ai ventilatori, ecc.). In altri casi il funzionamento è tipicamente intermittente, ad esempio: trapani, alza cristalli elettrici, ascensori. Talora il funzionamento è ciclico, ad esempio all'interno di una macchina automatica. Infine, ci sono situazioni in cui il moto è tipicamente vario, ad esempio trazione elettrica, robotica industriale. Inoltre, ci sono applicazioni in cui è necessario regolare la velocità del motore (ad esempio mandrino di un trapano, pompa, ventilatore, ecc.) ed altre in cui il motore va pilotato in posizione (ad esempio assi di un robot industriale o di una fresa a controllo numerico), più raramente si utilizza un controllo di coppia.

Talvolta la regolazione è grossolana (ad esempio ventilatore acceso/spento a una o due velocità), altre volte è richiesta una regolazione molto precisa (ad esempio macchina a controllo numerico). Per far fronte a questa enorme varietà di esigenze sono state sviluppate differenti tipologie di motori e di relativi sistemi di regolazione.

Nei decenni scorsi, prima che cominciasse lo sviluppo dell'elettronica di bassa ed alta potenza i sistemi di regolazione erano più semplici e ciascuno di essi dedicato a specifici campi applicativi. Lo sviluppo dell'elettronica analogica e digitale ha permesso la realizzazione di azionamenti più sofisticati nonché l'estensione del campo d'impiego dei motori esistenti. Solo per citare un esempio, consideriamo il classico motore asincrono. Nel passato era usato quasi esclusivamente per realizzare movimenti a velocità costante non regolata (marcia avanti, marcia indietro), con possibili variazioni di velocità di qualche punto percentuale in funzione del carico. L'introduzione degli inverter ha permesso di ottenere a costi ragionevoli la possibilità di regolare la velocità con continuità. La tecnica a controllo vettoriale consente un raffinato controllo del motore asincrono in coppia, in velocità ed in posizione.

A fronte di questa grande varietà di situazioni, il presente lavoro si concentra su una particolare categoria di motori elettrici, i **Motori Brushless SPM** (*surface permanent magnet*), essendo questi motori privi di spazzole e con eccitazione attraverso

magneti permanenti (montati sul rotore), la principale motivazione del loro sviluppo è stata la riduzione dei costi di manutenzione, la diminuzione delle dimensioni e l'aumento in termini della efficienza rispetto ai tradizionali **Motori DC con spazzole**. Inoltre, il basso costo di produzione, lo sviluppo di nuovi e migliori materiali e gli enormi progressi dell'elettronica di potenza hanno aperto la strada per la diffusione di questa tecnologia; un'impareggiabile densità di potenza è stato realizzato in seguito all'adozione di magneti permanenti con elementi chimici della serie dei lantanidi.

L'azionamento **AC Brushless** può distinguersi dall'azionamento **DC Brushless** poiché le correnti hanno una forma d'onda sinusoidale. Questa funzione consente di produrre una coppia con bassissimo contenuto armonico, rendendo l'azionamento **AC Brushless** una scelta naturale per le applicazioni ad alte prestazioni come ad esempio sistemi di posizionamento. Purtroppo, per ottenere correnti con forme d'onda sinusoidale, si deve utilizzare sensori di posizione ad alta risoluzione. I valori tipici di risoluzione che sono necessari sono pari a 10-12 bit per coppia polare. Tali sensori aumentano il costo delle unità e ridurre la loro affidabilità.

D'altra parte, per l'azionamento **DC Brushless** il basso costo deriva del fatto che ha soltanto bisogno di un trasduttore di posizione angolare economico a bassa risoluzione per raggiungere una corretta commutazione. Un tipico traduttore che viene utilizzato è l'encoder composto da sonde ad effetto Hall, perché è estremamente robusto e privo di manutenzione, dato che tipicamente si incolla sullo statore. Se questo tipo di trasduttore è destinato a essere utilizzato in un azionamento **AC Brushless**, è necessario utilizzare appropriati sistemi di elaborazione del segnale per estrarre le stime ad alta risoluzione di posizione e velocità angolare.

“Lo scopo di questa tesi” è quello di indagare l'adozione di un'encoder composto da 2 sonde ad effetto Hall di bassa risoluzione messe in quadratura in un azionamento **AC Brushless**. Molte applicazioni potrebbero trarre vantaggio da l'uso di questo azionamento: per esempio pompe a velocità variabile, il motore di una vasca di idromassaggio, per citarne alcuni.

I.2. Descrizioni dei capitoli

Nel capitolo 1 si presenta un riassunto sugli azionamenti elettrici dove si descrive la macchina sincrona a magneti permanenti (**SPM-surface permanent magnet** e **IPM-interior permanent magnet**), con particolare approccio negli azionamenti AC Brushless e DC Brushless. Si descrivono i componenti principali di un azionamento elettrico, i tipi di traduttori utilizzati per la misura della posizione angolare e di velocità angolare. Inoltre si riportano le caratteristiche più importanti degli azionamenti con motori brushless, specificamente l'azionamento AC Brushless e DC Brushless. Per ultimo, si fanno le conclusioni sugli azionamenti elettrici, con lo studio dei possibili miglioramenti in termini di risparmio economico ed aumento di prestazioni.

Il capitolo 2 descrive il procedimento per la stima della posizione e velocità attraverso un'encoder composto da 2 sonde ad effetto Hall a bassa risoluzione messe in quadratura: si utilizza un Vector Tracking Observer VTO (Osservatore ad inseguimento vettoriale). Poi si analizza i problemi di quantizzazione e gain scheduling prodotti dall'utilizzo di tale encoder.

Nel capitolo 3 si descrive il hardware e software del Digital Signal Processors (DSP) per l'implementazione dell'osservatore, nella parte del hardware si fa riferimento alla struttura, caratteristiche tecniche e periferiche utilizzati; nella parte del software si presenta l'algoritmo per l'implementazione dell'osservatore.

Il capitolo 4 presenta le simulazioni attraverso Matlab e risultati sperimentali ottenuti dall'implementazione dell'osservatore con il DSP; ulteriormente si fa un confronto tra i risultati delle simulazioni e quelli delle prove sperimentali sull'osservatore.

Per ultimo, si fa una breve sintesi del lavoro realizzato e le conclusioni generali dei risultati che ne sono ottenuti.

Capitolo 1

Riassunto sugli Azionamenti elettrici

1.1 Introduzione

Nel Capitolo 1 si presenta un riassunto sugli Azionamenti Elettrici; nel paragrafo 1.2 si descrive il Motore sincrono a magneti permanenti (**SPM-surface permanent magnet** e **IPM-interior permanent magnet**), con enfasi sugli azionamenti AC Brushless e DC Brushless. Le caratteristiche generali ed i componenti di un azionamento elettrico sono descritti nel paragrafo 1.3. Il paragrafo 1.4. compara i principali trasduttori di posizione e velocità utilizzati nei azionamenti elettrici. Il paragrafo 1.5. descrive le caratteristiche più importanti negli azionamenti con motori brushless, come si può modellare la macchina sincrona (equazioni descrittive, trasformazioni di variabili ed il bilancio energetico). Per ultimo, le conclusioni sugli azionamenti elettrici sono indicate nel paragrafo 1.6.

1.2 Motore sincrono a magneti permanenti

I **Motori sincroni a magnete permanente** [Azi¹], sono impiegati sempre più diffusamente in ambito industriale, specialmente nei servoazionamenti di piccola e media potenza. Essi sono essenzialmente destinati ad azionamenti ad elevate prestazioni, in cui le particolari specifiche giustifichino il loro costo che è solitamente elevato per la presenza di magneti permanenti di pregio nell'elemento mobile (rotore). La *conversione elettromeccanica* che essi attuano segue il principio di funzionamento dei *sistemi elettrodinamici* in cui però i conduttori su cui agiscono le forze sono collocati nella parte fissa (statore) ed il rotore viene posto in movimento per il principio fisico di reazione. Una rappresentazione schematica della struttura di un motore sincrono a magneti permanenti a due poli è mostrata in Fig. 1.1.

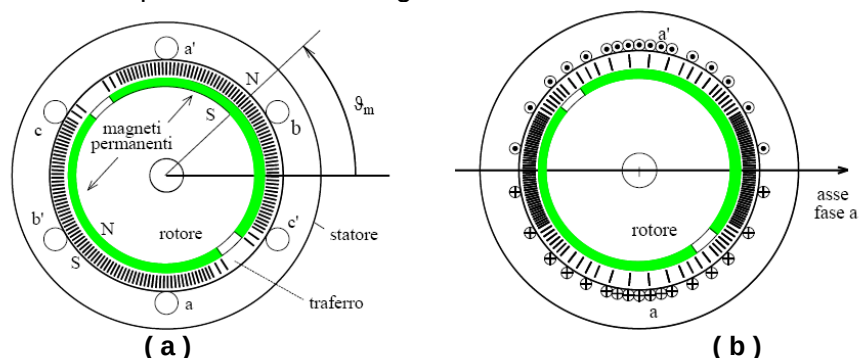


Fig. 1.1. Rappresentazione schematica di un Motore Sincrono a Magnet Permanenti a due poli [Azi¹]

(a) -. Induzione al traferro prodotta dal Magnete Permanente di rotore (trapezoidale)

(b) -. Induzione al traferro prodotta dall'avvolgimento statorico della fase a (sinusoidale).

Lo statore ed il rotore sono entrambi a forma di corona cilindrica di materiale ferromagnetico laminato e separati da un *traferro* in aria. Sul rotore trovano posto i magneti permanenti; dato che essi presentano generalmente una permeabilità magnetica differenziale molto simile a quella dell'aria, a seconda della loro disposizione e della forma del rotore si possono ottenere strutture di rotore isotrope o anisotrope dal punto di vista magnetico, che caratterizzano rispettivamente i **Motori Brushless SPM** (*surface permanent magnet*) e **Motori Brushless IPM** (*interior permanent magnet*). L'avvolgimento di statore è di tipo trifase; le tre fasi sono reciprocamente sfasate nello spazio di $2\pi/3$, e ciascuna fa capo ad una coppia di morsetti indicati con aa', bb', cc' in Fig. 1.1(a), attraverso i quali è possibile fornire loro alimentazione da una sorgente trifase esterna. I conduttori che compongono ciascuna fase (Fig. 1.1(b)) sono distribuiti lungo le cave statoriche ricavate secondo la direzione delle generatrici del cilindro di statore, omesse per chiarezza nel disegno. La stessa figura riporta, in (a), una rappresentazione schematica in cui ciascuna fase è simbolicamente rappresentata con una sola coppia di conduttori; si intende che l'asse di ogni fase sia la retta normale al piano che passa per ciascuna coppia di conduttori (Fig. 1.1 (b)).

I **Motori Brushless SPM** (*surface permanent magnet*) [SiS¹] possono essere praticamente considerati, da un punto di vista magnetico, dei motori isotropi con un elevato spessore di *traferro*, in quanto i magneti hanno una permeabilità relativa prossima ad 1 ($1.02 \div 1.2$); le induttanze di magnetizzazione secondo gli assi diretto ed in quadratura di rotore sono di conseguenza uguali. Inoltre, a causa dell'elevato *traferro*, l'induttanza sincrona è piccola e quindi gli effetti della reazione di armatura sono trascurabili.

I **Motori Brushless IPM** (*interior permanent magnet*) presentano una struttura meccanica più robusta, il che li rende particolarmente adatti ad applicazioni ad alta velocità, poiché i magneti sono fisicamente contenuti all'interno del rotore e protetti. Benché tali motori abbiano un rotore con una struttura geometrica isotropa, non possono essere considerati isotropi da un punto di vista del circuito magnetico.

I **Motori sincroni a magneti permanenti** si possono suddividere in due principali categorie:

- **DC Brushless**, aventi distribuzione trapezoidale (BLDM) dell'induzione al *traferro*;
- **AC Brushless**, aventi distribuzione sinusoidale (PMSM) dell'induzione al *traferro*.

La differenza è solo costruttiva e dipende dalla disposizione dei magneti e dalle distribuzioni degli avvolgimenti di statore: in sintesi, nel primo caso la forza elettromotrice indotta avrà una forma trapezoidale, nell'altro caso sinusoidale.

Le tecniche di controllo saranno diverse nei due casi: in particolare nel caso trapezoidale verranno alimentate due fasi alla volta, mentre nel caso sinusoidale verranno alimentate le tre fasi contemporaneamente.

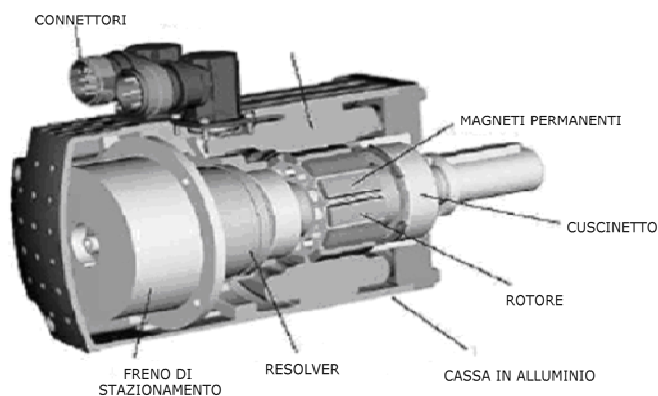
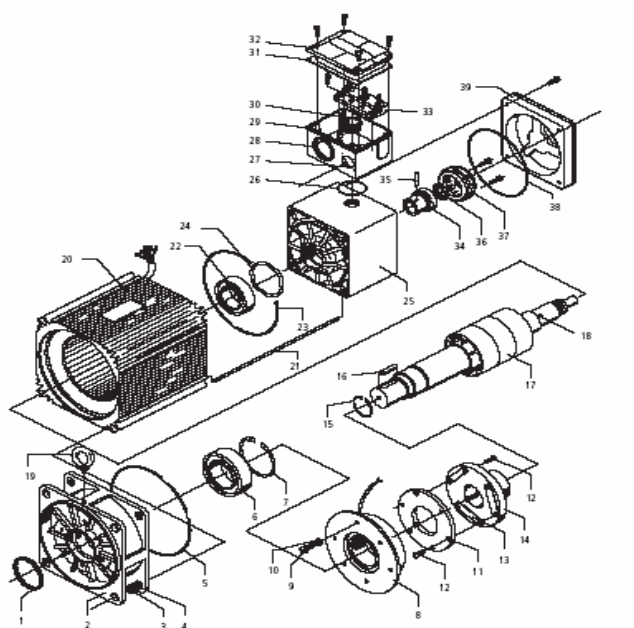


Fig. 1.2. Sezione di un Motore Brushless [SiS¹]



Descrizione parti

- | | |
|----------------------------|------------------------------------|
| 1 Paraolio | 20 Gruppo carcassa statore |
| 2 Calotta anteriore | 21 Tirante |
| 3 Dado | 22 Cuscinetto posteriore |
| 4 Rondella | 23 Anello di tenuta statica |
| 5 Anello di tenuta statica | 24 Molla di precarico |
| 6 Cuscinetto anteriore | 25 Calotta posteriore |
| 7 Anello elastico | 26 Anello di tenuta statica |
| 8 Freno | 27 Foro connettore segnali |
| 9 Vite | 28 Foro PG potenza |
| 10 Rondella | 29 Coprimorsettiera |
| 11 Anello d'armatura | 30 Ghiera di fissaggio |
| 12 Vite | 31 Guarnizione |
| 13 Molla a lamina | 32 Coperchio |
| 14 Flangia | 33 Morsettiera |
| 15 Anello elastico | 34 Rotore generatore tachimetrico |
| 16 Chiavetta | 35 Spina |
| 17 Gruppo rotore | 36 Ghiera di fissaggio |
| 18 Foro spina | 37 Statore generatore tachimetrico |
| 19 Golfare | 38 Anello di tenuta statica |
| | 39 Coperchio posteriore |

Fig. 1.3. Vista parti di un Motore Brushless [SiS¹]

1.3. Gli Azionamenti Elettrici

1.3.1. Caratteristiche generali di un Azionamento Elettrico

Nel campo degli azionamenti [Azi²] per macchine utensili una distinzione che viene solitamente introdotta è quella tra assi e mandrini indicando con i primi quegli azionamenti destinati esclusivamente ai moti di avanzamento, con i secondi quelli responsabili dei moti di lavoro dell'utensile o del pezzo a seconda del tipo di macchina.

Si passa quindi ad analizzare brevemente i requisiti di un azionamento asse: il suo compito è essenzialmente quello di portare in rotazione un albero ad una determinata velocità, imposta da un opportuno riferimento, indipendentemente, entro determinati limiti, dalla coppia resistente e quindi dalla coppia motrice erogata dal motore; questo compito deve poter essere svolto rispettando alcuni importanti requisiti:

- Totale bidirezionalità dell'azionamento con zona morta praticamente nulla intorno allo zero di velocità, sia in condizioni statiche che dinamiche;
- Il rapporto tra velocità massima e minima regolabile deve essere indicativamente maggiore di 10 con coppia nominale e, passando da vuoto a carico nominale, la velocità non deve diminuire più di 1/10 della velocità massima.

Tali caratteristiche possono essere riassunte dicendo che l'azionamento deve possedere una elevata rigidità statica, intendendo per rigidità statica appunto il rapporto tra coppia esterna applicata e variazione di velocità da essa prodotta.

Altro elemento fondamentale di giudizio tecnico è la banda passante dell'anello di velocità: per un asse di elevate prestazioni è richiesto che la banda passante in fase non sia inferiore ai 40 Hz (cioè la frequenza alla quale lo sfasamento supera i 45°, sia superiore ai 40 Hz) mentre in ampiezza (frequenza in corrispondenza della quale si ha un'attenuazione di 3 dB) non sia inferiore ai 70 Hz. Tali caratteristiche, valide nel funzionamento attorno allo zero di velocità, devono mantenersi entro un margine del 20% rispetto a quanto definito sopra anche in condizioni di lavoro con "carico inerziale" (inteso convenzionalmente come un carico pari alla metà dell'inerzia del motore) o con un "carico di coppia nominale" (convenzionalmente definito come una coppia pari alla metà di quella nominale). L'ampiezza della banda passante dell'anello di velocità può dunque essere assunta come indice della rigidità dinamica dell'azionamento.

Infine l'accelerazione fornita dal motore in presenza di un carico inerziale o di coppia definiti come sopra deve essere elevata, indicativamente dell'ordine dei 100 rad/s²; per contro le velocità massime richieste non sono in genere molto elevate, aggirandosi attorno ai 200 rad/s.

Lo scopo di questi requisiti, peraltro strettamente correlati fra loro, è quello di poter ottenere, con l'ausilio di catene cinematiche rigide, un controllo di posizione particolarmente accurato al fine di raggiungere elevati gradi di finitura del pezzo in lavorazione.

1.3.2. Struttura di un Azionamento Elettrico

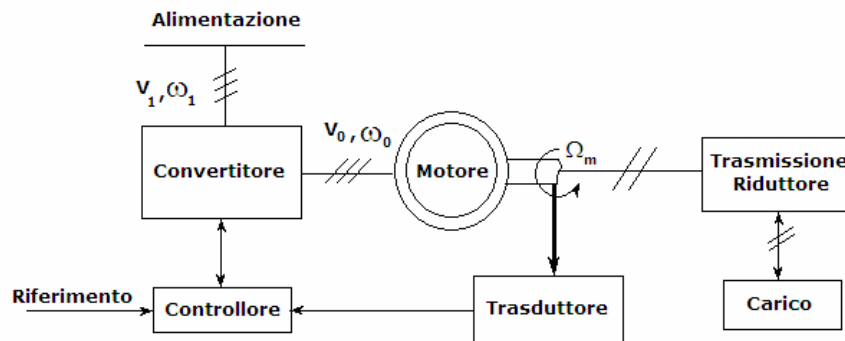


Fig. 1.4. Struttura di un Azionamento Elettrico

La Figura 1.4, indica i componenti principali di un azionamento elettrico [SiS¹] che sono:

1.3.2.1. Motore

L'elemento base è il motore, collegato a monte elettricamente alla rete elettrica ed a valle meccanicamente al carico, ossia al dispositivo meccanico che si vuole azionare (ad esempio il mandrino di una macchina utensile, un sistema di movimentazione, una pompa, ecc.)

1.3.2.2. Convertitore

Fra il motore e la rete di alimentazione si possono trovare i seguenti componenti elettrici o elettronici:

- **Adattatore delle grandezze elettriche** (ad es. un trasformatore, se la tensione richiesta dal motore differisce da quella di rete, oppure un raddrizzatore se il motore funziona in corrente continua). Se possibile si tende a ridurre al minimo la presenza di tali dispositivi.
- **Convertitore statico**, converte le caratteristiche elettriche della potenza di alimentazione a quelle richieste dal motore e spesso consente un controllo in posizione, velocità o coppia del carico. Il variatore elettronico è, normalmente, costituito da un convertitore statico pilotato. Un convertitore di questo tipo comprende due sezioni: quella di comando e quella di potenza. La prima (sempre più diffusamente del tipo a microprocessori) serve per decidere con quale legge far variare tensione e corrente di alimentazione del motore per far sì che esso si muova come richiesto. La parte di potenza si occupa della effettiva erogazione di tensione e corrente. Il convertitore è spesso impropriamente chiamato "azionamento", termine che invece definisce l'intero sistema di movimentazione. Il convertitore si dice "statico" perché non esistono parti in movimento. In alcuni casi l'alimentazione è a batteria.

L'alimentazione dei motori sincroni si realizza utilizzando un dispositivo chiamato **Inverter**. Si tratta di un convertitore statico di energia elettrica, più precisamente di un raddrizzatore e di un convertitore DC-AC. Esso infatti converte l'energia sotto forma continua in alternata. La funzione

dell'inverter, all'interno di un azionamento è di controllare l'alimentazione del motore. Nel caso più generale, infatti, gli inverter consentono di variare l'ampiezza e/o la frequenza della tensione o della corrente di alimentazione dei motori. A seconda della grandezza controllata l'inverter viene chiamato di tensione (VSI) o di corrente (CSI).

L'inverter di tensione controllato *ad onda quadra*, è alimentato da una tensione continua e permette di ottenere in uscita una tensione regolabile in frequenza. L'ampiezza della tensione di uscita, invece, non è regolabile.

Un controllo della ampiezza di tensione di uscita si ottiene per mezzo del controllo con la tecnica PWM. Negli inverter che utilizzano tale tecnica è quindi possibile regolare sia l'ampiezza che la frequenza della tensione di uscita. La PWM, oltre a fornire un controllo sull'ampiezza della tensione, permette di ridurre l'effetto delle armoniche diverse dalla fondamentale, che rappresentano dei disturbi da eliminare. Sfruttando l'azione filtrante esercitata dal carico, la PWM consente di ottenere una corrente praticamente sinusoidale.

La differenza sostanziale tra i vari tipi di convertitori non è nella sezione di potenza bensì nel modo in cui si comandano i transistor. La tecnica utilizzata nel PWM deriva da quella usata nell'ambiente delle telecomunicazioni, e consiste nel prendere una portante ad alta frequenza (tipicamente $10 \div 20$ KHz) di forma triangolare ed una modulante di forma sinusoidale (si parla di PWM sinusoidale) o quadrata (PWM quadrata); queste due forme d'onda vengono inviate ad un comparatore a due livelli, il quale fornisce in uscita un segnale modulato

1.3.2.3. Trasmissione

Il motore è normalmente collegato al carico tramite alcuni dei seguenti organi meccanici:

- **Adattatore meccanico di velocità**

In pratica un riduttore di velocità, dato che di solito le velocità ottimali per il progetto dei motori (spesso dai 1.000 ai 6.000 giri/min) sono superiori a quelle utili per il carico.

- **Organo di trasformazione del moto.**

Anche se esistono motori lineari, lo stesso movimento lo si può ottenere ad esempio con un dispositivo costituito da una coppia vite/madrevite o da una coppia pignone/cremagliera o da una trasmissione con cinghia (spesso cinghia dentata). Tale dispositivo funziona anche da "riduttore", in quanto permette di scegliere, entro certi limiti, il rapporto tra avanzamento del carico e rotazione del motore, sicché se il motore non è troppo veloce si può eliminare il riduttore vero e proprio. Quando è necessario produrre moto vario possono essere impiegati altri componenti, ad esempio camme o sistemi articolati.

1.3.2.4. Controllore

Il controllore è un dispositivo opzionale presente nei casi in cui sia necessario regolare la velocità di rotazione del motore o fargli assumere una legge di moto prefissata. Il controllore è spesso costituito da un sistema a microprocessore. Il controllore può ricevere informazioni sullo stato del motore da appositi dispositivi di controreazione. Tra i più comuni tipi di controllore troviamo quelli detti controlli numerici (computerizzati) detti CN o CNC e i PLC (Programmable Logic Circuit). Entrambi sono basati su sistemi a microprocessore. I controlli numerici permettono la gestione di sistemi complessi che richiedono una perfetta sincronizzazione di vari attuatori. I PLC sono utilizzati nei casi più semplici, quando ci siano pochi calcoli da effettuare ed il coordinamento consista per la massima parte in una verifica di consensi e valori dello stato di sensori, comandi o attuatori. Sistemi complessi integrano CN e PLC.

1.3.2.5. Sistema di Controreazione

Quando è necessario pilotare il motore con precisione possono essere utilizzati uno o più trasduttori per rilevare la posizione, la velocità o la coppia trasmessa dal motore. I segnali letti da questi trasduttori vengono inviati al controllore e/o al convertitore, che tenendo conto di queste informazioni riescono a pilotare il motore correttamente.

1.4. Trasduttori di posizione angolare e velocità

1.4.1. Risoluzione, Accuratezza assoluta ed Accuratezza differenziale

La prestazione di un trasduttore di posizione (encoder) è generalmente definita per [Ken¹] tramite la risoluzione nei rispettivi data sheets. La risoluzione però, non può descrivere completamente la caratteristica di controllo di un encoder di posizione, specialmente quando si usa lo stesso trasduttore per il controllo di velocità. Per questo motivo si aggiungono due dati in più che sono necessari, l'accuratezza assoluta e l'accuratezza differenziale.

La risoluzione “ r ” non è più che una misura del numero dei valori reali differenti che il trasduttore può distinguere uno dell'altro. Nel caso di un encoder di posizione è il numero di o la distanza tra diverse posizione che possono essere identificate, nel caso di un encoder di velocità è il numero di o la distanza tra diversi valori di velocità che possono essere identificati. Di conseguenza, la risoluzione di un encoder di posizione senza interpolazione coincide con il numero “ n ” di poli, denti, linee o segmenti dell'encoder stesso, come si indica nell'equazione 1.1:

$$(1.1) \quad r = n$$

In questo caso la definizione di una risoluzione differenziale r' non è di grande aiuto, perché il numero di linee, segmenti, ecc., sono indipendenti della posizione di riferimento o qualcun'altra condizione di operazione dell'encoder (come si vede nell'eq. 1.2):

$$(1.2) \quad r' = \frac{dr}{dx} = 0$$

L'accuratezza assoluta "a" di un trasduttore di posizione è definito come la differenza tra la posizione di riferimento $X_{rif,i}$ e la posizione $X_{reale,i}$, tutte due sono indicati dal trasduttore. L'equazione 1.3 descrive l'accuratezza assoluta rispetto alla distanza ideale tra due posizioni identificate, mediata lungo una gamma di operazione (di solito 1 rotazione).

$$(1.3) \quad a = \frac{|X_{rif,i} - X_{reale,i}|}{\sum_i X_{reale,i} / n} = n \cdot \frac{|X_{rif,i} - X_{reale,i}|}{\sum_i X_{reale,i}}$$

L'accuratezza differenziale a' di un trasduttore di posizione è definito come la differenza tra la distanza reale tra due posizioni vicine e la distanza ideale tra due posizioni successive (questo è uguale a fare tutta una gamma di operazioni divisa per il numero di posizioni identificate). L'equazione 1.4 definisce a' come una quantità relativa (simile a "a" nell'equazione 1.3):

$$(1.4) \quad a' = \frac{|X_{reale,i} - X_{reale,i-1} - \sum_i X_{reale,i} / n|}{\sum_i X_{reale,i} / n} = n \cdot \frac{|X_{reale,i} - X_{reale,i-1} - 1|}{\sum_i X_{reale,i}}$$

Non è automaticamente certo che una grande risoluzione vuol dire che si ha una grande accuratezza o viceversa. Un esempio tipico per l'encoder di posizione di bassissima risoluzione (per esempio 18 per rivoluzione) e grande accuratezza assoluta (per esempio $0,3^\circ$) sarebbe l'encoder a commutazione nel cosiddetto Azionamento DC Brushless (BLDC), la risoluzione è uguale al numero di commutazioni per rivoluzione, mentre l'accuratezza assoluta doveva essere molto alta per evitare un ripple di coppia a causa della commutazione stessa.

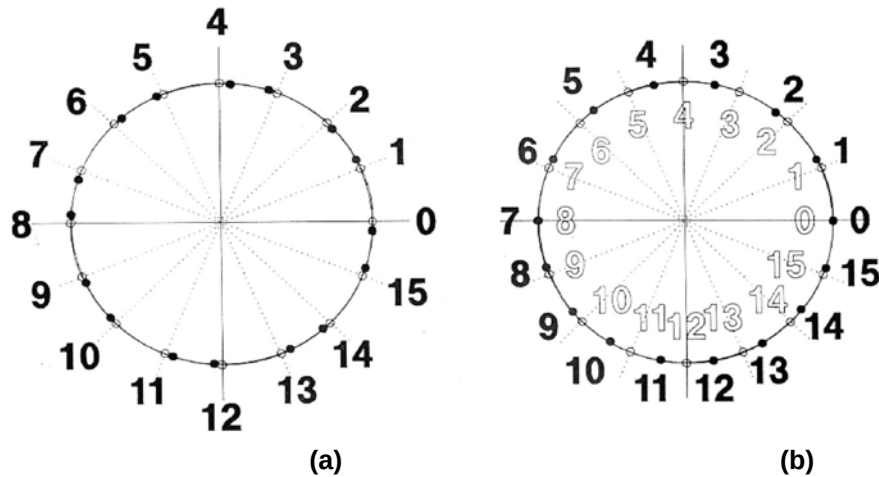


Fig. 1.5 . Accuratezza assoluta (a) e differenziale (b) [Ken¹]

La figura 1.5 mostra la differenza tra l'accuratezza assoluta e l'accuratezza differenziale in maniera grafica. Questa sarebbe la situazione di un encoder con una risoluzione di 16, l'encoder è capace di distinguere tra 16 differenti posizioni. Per tutte due figure, la posizione ideale è segnata con i puntini bianchi e poi con linee segmentate che partono dal centro del cerchio; nel caso della posizione reale si segnano con i puntini neri.

Nel caso (a) della fig. 1.5, l'encoder riconosce la posizione rispettiva poco prima o poco dopo della posizione ideale. Nel caso (b), le distanze tra le posizioni reali riconosciute sono leggermente più lunghe nella parte superiore del cerchio (sopra la metà) e leggermente più corta nella parte inferiore del cerchio (sotto la metà). Dato che i valori reali nell'anello di controllo di posizione dipende della posizione dell'encoder, la sua accuratezza assoluta è più importante per l'accuratezza dell'anello di controllo di posizione stesso. Invece l'accuratezza differenziale è più importante per l'anello di controllo di velocità, dato che i valori reali per il controllo di velocità sono ottenuti per differenziazione dell'uscita dell'encoder di posizione.

Purtroppo i data sheets di solito per gli encoder contengono solo informazione della risoluzione, raramente informazione dell'accuratezza assoluta e difficilmente informazione dell'accuratezza differenziale.

1.4.2. Trasduttori di posizione angolare

I trasduttori di posizione angolare di tipo digitale [Azi¹] sono dispositivi nei quali la posizione angolare è quantizzata, cioè l'angolo di rotazione dell'albero mobile ($0 \div 360^\circ$) è suddiviso in un numero discreto di parti a ciascuna delle quali viene fatto corrispondere un segnale digitale. Esistono principalmente due tipi di trasduttori angolari con uscita digitale: l'encoder incrementale e l'encoder assoluto. Normalmente, dal segnale di posizione si può ricavare anche un segnale di velocità.

1.4.2.1. Encoder incrementale

L'encoder incrementale è definito per [Azi¹] e [Elt¹] come un trasduttore rotante che genera uno o due treni d'impulsi (versione mono o bidirezionale), sinusoidali o quadri, caratterizzati da un certo numero di impulsi per ogni angolo giro dell'albero (risoluzione dell'encoder) e sfasati 90° elettrici tra di loro, che in genere sono "Canale A" e "Canale B". Con la lettura di un canale solo si dispone della informazione corrispondente alla velocità di rotazione, mentre se prendiamo anche il segnale "B" è possibile sapere il senso di rotazione che dipende della sequenza dei dati che producono i entrambi i segnali. In aggiunta, si può generare per ogni giro un impulso supplementare detto di zero o di riferimento, che definisce la posizione di zero (0) assoluto dell'asse dell'encoder. In fig. 1.6 è illustrata la schematizzazione di un encoder incrementale.

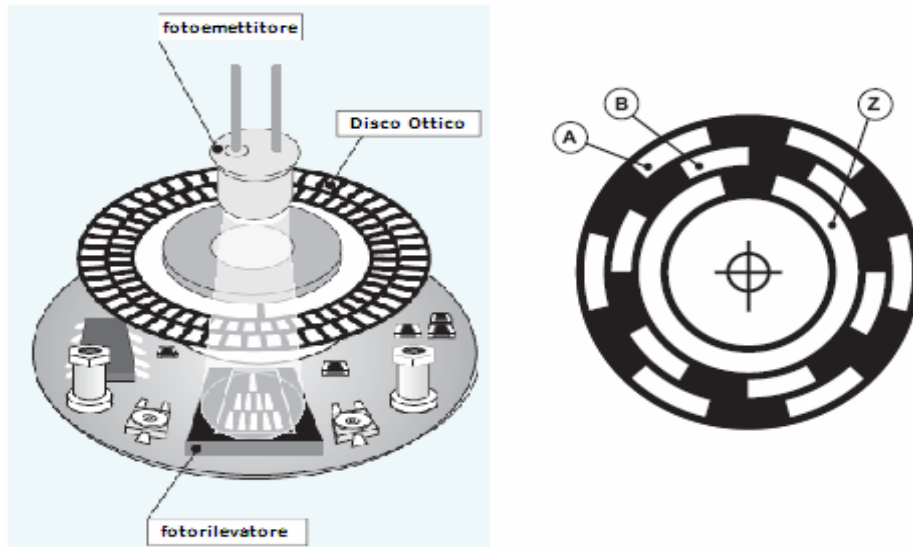


Fig. 1.6. Schematizzazione di un encoder incrementale [El¹]

I segnali generati per l'encoder incrementale ad onda quadra e sinusoidale, sono riportati in fig. 1.7.

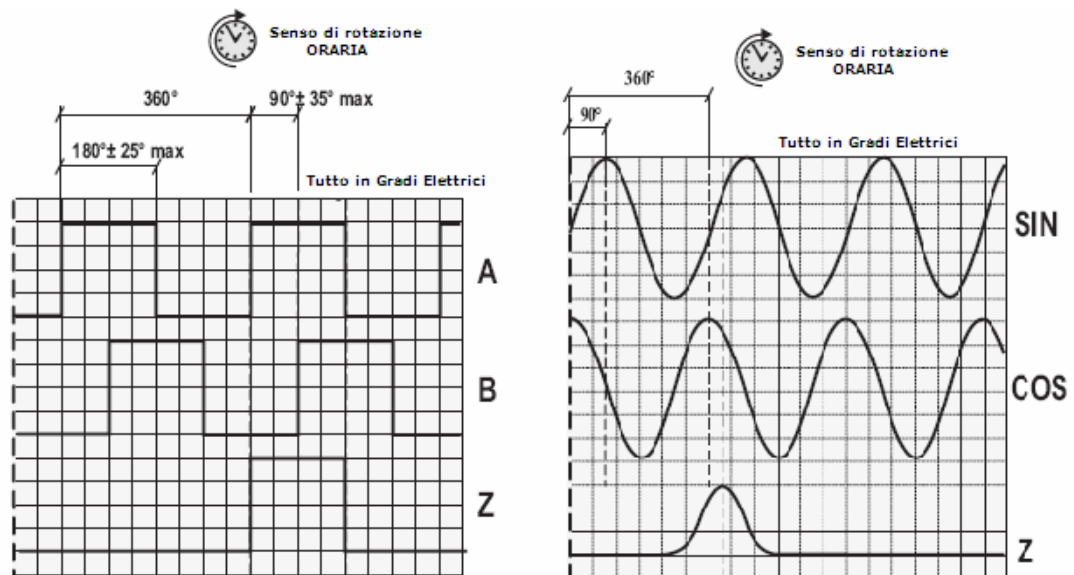


Fig. 1.7. Uscita per l'encoder incrementale ad onda quadra e sinusoidale [El¹]

La precisione di un encoder incrementale dipende di fattori meccanici ed elettrici, come per esempio: l'errore di divisione del reticolo, l'eccentricità del disco, l'errore introdotto per l'elettronica di lettura, imprecisioni di tipo ottico, ecc.

In particolare, per abbinamento a convertitori per servomotori sincroni, gli encoder sinusoidali forniscono ulteriori due segnali seno-coseno (detti di commutazione e caratterizzati da un periodo al giro) per il rilevamento assoluto della posizione rotorica del motore all'atto dell'inserzione dell'alimentazione; dopo l'inserzione, l'elaborazione interessa i segnali incrementali, trattandoli con tecniche d'interpolazione per ottenere risoluzioni fino ad alcune decine di milioni di passi al giro. All'albero, cavo o sporgente, dell'encoder è solidamente fissato un disco di materiale trasparente sul quale sono fotoincise delle zone opache. Il

principio di funzionamento è analogo a quello dell'encoder assoluto, solo che ora le zone opache si alternano a quelle trasparenti senza alcuna codifica. La superficie del disco è illuminata da fotodiodi, in modo che le zone opache in movimento intercettino a tratti il fascio luminoso della sorgente. Dalla parte opposta ai fotodiodi sono collocati i fototransistori di rilevazione; questi trasformano il segnale luminoso modulato in segnale elettrico con forma d'onda quadra o sinusoidale. L'accoppiamento dell'albero encoder all'albero che lo conduce avviene per mezzo di un giunto elastico in senso assiale e radiale, rigido in senso torsionale. Per non guastare il disco ottico interno, sono rigorosamente da evitarsi urti dell'encoder e lavorazioni, pressioni e flessioni del relativo albero.

I dati caratteristici degli encoder con uscita ad onda quadra sono riassunti di seguito. In particolare gli encoder sinusoidali con segnali incrementali sono contraddistinti da:

- frequenza operativa: ~ 500 kHz
- grado di protezione: IP20
- temperatura operativa: -10 °C ÷ 120 °C
- massima velocità di rotazione: 12000 giri/min
- stadio elettronico d'uscita: line driver
- albero: cavo.

E' da notare che l'estensione del range operativo di temperatura a 120 °C è particolarmente importante per i servomotori privi di ventilazione forzata. Per quanto riguarda il cavo di connessione all'encoder valgono le seguenti considerazioni:

- è di tipo schermato, con schermo normalmente connesso a terra dal lato opposto all'encoder;
- va posato in modo separato da cavi di potenza o d'alimentazione di carichi quali elettrovalvole, teleruttori, ecc;
- ha una sezione di 0,25 mm² per i conduttori di segnale e di almeno 0,5 mm² per quelli d'alimentazione (+ Vcc e 0V). Questo è specialmente importante per gli encoder a 5 V, dato che una sezione di 0,5 mm² crea una caduta di tensione di circa 0,1 V ogni 10 metri con 100 mA di corrente ($\rho_{Cu}=17,21 \cdot 10^{-9} \Omega \cdot m$): con un cavo di circa 30 metri la parte interna dell'encoder (TTL) opera quasi alla minima tensione consentita (4,75 V).

1.4.2.2. Encoder assoluto

In questo tipo di trasduttore [Azi¹] l'uscita è rappresentata da una combinazione di livelli logici (0 e 1) quanti sono i bit che formano il segnale d'uscita. In fig. 1.8 è illustrata la schematizzazione di un encoder assoluto.

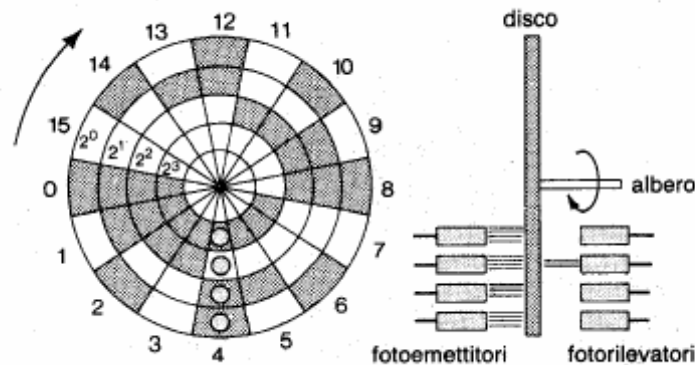


Fig. 1.8. Schematizzazione di un encoder assoluto [Azi¹]

La parte mobile del dispositivo è costituita da un disco che può ruotare attorno ad un asse calettato sulla macchina di cui si vuole rilevare la rotazione angolare. Il disco è suddiviso in un numero di settori pari a 2^n essendo n il numero di bit del segnale d'uscita (nell'esempio della figura i bit d'uscita sono $n = 4$, per cui i settori sono $2^4 = 16$). Evidentemente quanto maggiore è il numero dei settori, tanto maggiore è la risoluzione nella misura dell'angolo. Il disco è ulteriormente suddiviso in n corone circolari. Nel reticolo determinato dalle intersezioni delle linee radiali e dei cerchi concentrici si diversificano opportunamente delle zone (scure e chiare in fig. 1.8), che permettono la codifica binaria dell'angolo. La lettura può essere effettuata con mezzi ottici, in tal caso le zone scure sono rese opache e quelle chiare trasparenti. La parte fissa del dispositivo di lettura consiste in n sorgenti luminose, ognuna delle quali proietta sul disco un fascio di luce, e di un sistema di n rivelatori allineati con le sorgenti di luce e in posizione opposta al disco. A ciascuna corona circolare viene fatta corrispondere una potenza di 2, *crescente dal bordo esterno verso il centro*. Nel caso schematizzato nella fig. 1.8 si hanno pertanto quattro diversi valori corrispondenti alle corone indicate: 2^0 , 2^1 , 2^2 , 2^3 .

Fra i terminali di ogni fotorivelatore si preleva il livello logico 0 se il dispositivo non riceve luce e il livello 1 se la riceve: così, ad esempio, se tra i fotoemittitori e i foto rivelatori è interposto il settore 0, i livelli in uscita dai quattro fotodiodi risultano 0000; facendo ora ruotare il disco in senso orario, quando fra i fasci luminosi e il sistema fotosensibile è interposto il settore 1 la configurazione dei livelli d'uscita risulta 0001. Esiste pertanto una corrispondenza univoca fra il settore del disco, che si trova interposto fra le sorgenti di luce e il sistema rivelatore, e la configurazione dei livelli logici d'uscita. La lettura ottica permette dunque di codificare la posizione angolare in codice binario puro. In pratica però il codice binario puro non è usato nei codificatori di precisione poiché può portare a degli errori di lettura, in quanto nel passaggio da un numero a quello immediatamente successivo si possono presentare variazioni simultanee di più bit. Ad esempio, i due numeri decimali contigui 7 e 8 hanno la rappresentazione binaria: 0111 e 1000. Un errore di allineamento del fascio potrebbe portare,

nell'esempio indicato, alla lettura del numero 1111 rappresentativo del numero decimale 15. Il codice che viene usato per evitare questo tipo di errore, e sulla base del quale viene eseguita quindi la configurazione delle zone opache e trasparenti del disco, è tipicamente il codice Gray. Quest'ultimo è caratterizzato dal fatto che nel passaggio fra due numeri decimali consecutivi si ha sempre la variazione di un solo bit. La tabella Tab. 1 riporta il codice Gray per un sistema a 4 bit.

Tab. 1. Codice Binario e Gray per un sistema a 4 bit

Binario	Gray
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

E' possibile realizzare anche un sistema a lettura elettrica anziché ottica. Questo tipo di lettura può essere ottenuto realizzando le parti grigie del disco con materiale isolante, quelle bianche con materiale conduttore. Mettendo in tensione le parti conduttrici del disco, un sistema costituito da tante spazzole quante sono le corone circolari può prelevare la tensione lungo le corone stesse.

1.4.2.3. Resolver

Il resolver [Azi¹] è un trasduttore rotante con due avvolgimenti statorici, sfasati tra loro di 90° elettrici, ed un avvolgimento rotorico (fig. 1.9). Eccitando l'avvolgimento rotorico con una tensione alternata, in un avvolgimento statorico si induce una tensione d'ampiezza proporzionale al seno ($S_2 - S_4$) e nell'altro proporzionale al coseno ($S_1 - S_3$) dell'angolo di rotazione del rotore rispetto allo statore (fig. 1.10).

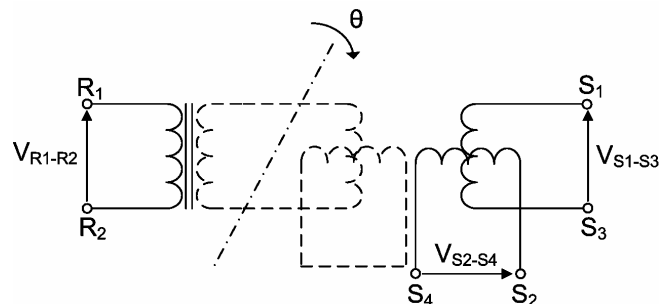


Fig. 1.9. Resolver eccitato attraverso l'avvolgimento rotorico ($R_1 - R_3$) [DeD¹]

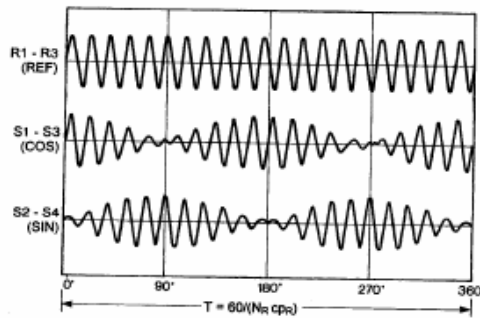


Fig. 1.10. Tensione indotta negli avvolgimenti statorici ($S_1 - S_3$ e $S_2 - S_4$) [Azi¹]

Le seguenti equazioni descrivono quello detto precedentemente, con K essendo la costante di proporzione di trasformazione di rotazione [DeD¹]:

$$(1.5) \quad V_{R1-R2} = V \sin(\omega_e t)$$

$$(1.6) \quad V_{S1-S3} = K V_{R1-R2} \cos(\theta)$$

$$(1.7) \quad V_{S2-S4} = K V_{R1-R2} \sin(\theta)$$

Avendo ottenuto tensioni che contengono informazione sulla posizione sia in fase sia in ampiezza, è necessario estrarre questa informazione per poi convertirla dal tipo analogico al dominio digitale. Per fare questo si usa il resolver eccitato attraverso l'avvolgimento rotorico insieme al Resolver to Digital Converter RDC (Convertitore digitale per il resolver).

Nella fig. 1.11 si vede la struttura tipica del RDC. Le due uscite del resolver sono passate per un moltiplicatore di seno (Sine multiplier) ed uno di coseno (Cosine multiplier), questi moltiplicatori hanno incorporati tabelle con i valori memorizzati che sono richiamate per dare il valore desiderato del seno o del coseno dell'angolo senza essere calcolati; i due segnali risultanti si sottraggono uno dall'altro in un amplificatore di errore per produrre un segnale di errore AC, $V \sin(\omega_e t) [\sin(\theta - \phi)]$, per essere inviato al Detector. La funzione del Detector è demodulare il segnale di errore AC usando come riferimento la tensione nel rotore del resolver (V_{R1-R2}) e produrre un segnale di errore DC proporzionale a $\sin(\theta - \phi)$. Quando $\phi \rightarrow \theta$, $\sin(\theta - \phi) \approx (\theta - \phi)$, si ha una deviazione angolare piccola, dunque il segnale di errore è proporzionale all'errore di riferimento. Il segnale di errore DC è integrato ed inviato al Voltage Controlled Oscillator VCO (Oscillatore controllato per tensione). Il VCO provoca un conteggio nel up/down counter nella direzione giusta per ridurre a zero l'errore di fase ad inseguimento, nel giro di un conteggio. I Latches sono usati per memorizzare e poi trasferire il valore di posizione calcolato all'esterno senza interrompere l'anello di inseguimento. Un RDC è in pratica un Phase-Locked Loops (PLL) di secondo ordine, poiché possiede due integratori: il primo è proprio un integratore del segnale e l'altro è il VCO. La presenza di un doppio integratore è necessario per portare a zero l'errore di posizione ad inseguimento per il motore a velocità costante.

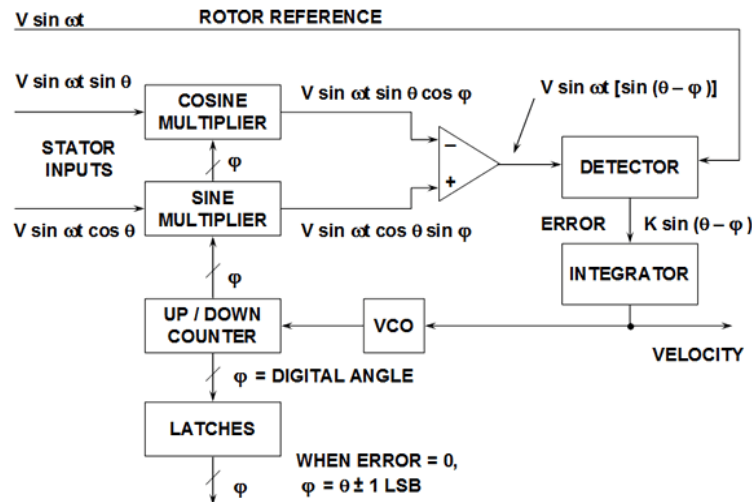


Fig. 1.11. Struttura del convertitore digitale per il resolver (RDC) [DeD¹]

L'uscita dell'integratore determina una stima della velocità angolare, per poi essere portata fuori del RDC dove quest'uscita può essere utilizzata in un anello di velocità. Nella tecnologia di punta, l'accuratezza di un RDC è circa 2 archi minuti. L'accuratezza totale del sistema composto da Resolver e RDC è uguale alla somma delle singole accurattezze e può arrivare ad un valore minimo di 7 archi minuti approssimativamente.

Il vantaggio più importante per scegliere questa tecnologia è quella di misurare la posizione assoluta in ogni istante, incluso l'avviamento della macchina, per questo motivo è molto diffuso in applicazioni di trazione elettrica. Un'altra ragione è quella che il resolver a livello meccanico è molto robusto ed immune ad inquinamento ambientale. Il RDC offre un alto grado di immunità al rumore a causa della doppia integrazione della segnale di errore; inoltre il rumore a banda larga è bloccato dal detector purché non ecciti la sua frequenza.

Gli svantaggi possono essere di tipo tecnico ed economico. La prestazione del RDC viene limitata per il range di campionamento del demodulatore il quale deve essere campionato alla frequenza di rotazione del rotore; vuol dire che i contenuti dinamici devono essere inferiore alla metà della frequenza di eccitazione. In più, la frequenza di eccitazione non è sincronizzata con il rango di campionamento di controllo di moto, dunque possono presentarsi errori di moto durante l'accelerazione. Il resolver è una struttura complessa e quindi è piuttosto costoso, questo è dovuto alla necessità di utilizzare un RDC.

1.4.2.4. Sonde ad effetto Hall

Principio dell'effetto Hall [Hal¹]: quando un conduttore è attraversato da una corrente e da un flusso magnetico nella direzione perpendicolare a questi si sviluppa una differenza di potenziale.

Questo è provocato dalla forza di Lorenz $F = (qv) \times B$ che spinge le cariche libere verso un lato del conduttore. Queste si accumulano su di una faccia del campione fino a che il campo elettrico ad esse associato è sufficientemente grande da opporsi alla forza del campo magnetico. La differenza di potenziale tra i lati del campione (tensione di Hall, V_H) è quindi dipendente dall'intensità del

campo magnetico in cui è immerso il conduttore. La misura dell'intensità del campo magnetico B viene pertanto ricondotta alla misura della tensione di Hall.

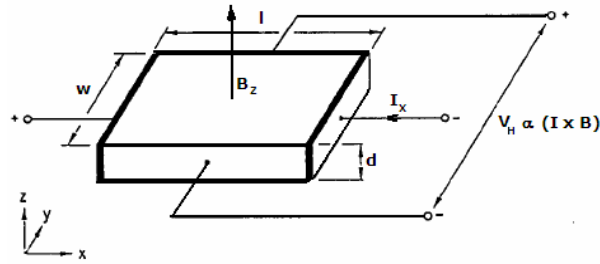


Fig. 1. 12. Principio base delle sonde ad effetto Hall

La tensione di Hall (V_H) è definita come [Hal²]:

$$(1.8) \quad \begin{cases} F = -q(v_x \times B_z) \Rightarrow E_H = v_x B_z \\ \text{La densità di corrente vale: } J_x = -q n v_x \text{ da cui: } E_H = -\frac{J_x B_z}{q n} \\ E_H w = -\frac{J_x B_z}{q n} \frac{w d}{d} \Rightarrow V_H = -\frac{I_x B_z}{q n d} \Rightarrow V_H = \frac{R_H}{d} I_x B_z \\ R_H = -\frac{1}{q n} \Rightarrow \text{Costante di Hall} \end{cases}$$

dove,

V_H : tensione di Hall in Volt

R_H : costante di Hall (dipende dal materiale)

d : è lo spessore dell'elemento

I_x : corrente che attraversa l'elemento in Amp

B_z : campo magnetico incidente nell'elemento

E_H : campo elettrico di Hall, lungo la direzione del moto

q : carica elettrica puntiforme

n : numero di particelle per unità di volume

v_x : velocità con cui si muove la carica lungo l'asse x

La tensione di Hall (V_H) è molto piccola, circa i 30 μV in presenza di una densità di flusso di campo magnetico di 1G (1G = 10^{-4} T) [DeD¹].

1.4.2.4.1. Sonda ad effetto Hall a misura diretta

Si considera un circuito magnetico [Hal¹] (fig. 1.13) costituito da un nucleo toroidale aperto, di ferrite o di altro materiale ferromagnetico. Nell'apertura è alloggiato il sensore ad effetto-Hall sul quale il nucleo toroidale concentra il campo magnetico generato dalla corrente elettrica incognita I_M (corrente che si vuole misurare).

Sul sensore Hall è fatta passare una piccola corrente di riferimento I_{REF} che, accoppiandosi con il campo magnetico genera per effetto Hall una tensione V_{OUT} proporzionale alla corrente I_M .

$$(1.9) \quad V_{OUT} = (K \cdot I_{REF}) \cdot I_M$$

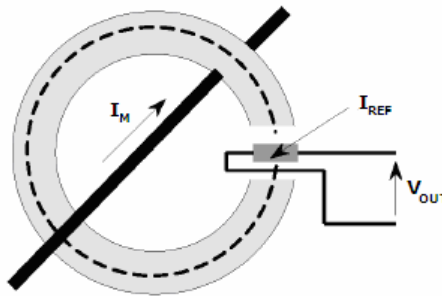


Fig. 1.13. Circuito magnetico per la sonda ad effetto Hall a misura diretta [Hal¹]

1.4.2.4.2. Sonda ad effetto Hall a compensazione di campo

Siccome [Hal²] il sensore ad effetto Hall risulta avere un buon comportamento lineare solo per bassi valori di induzione magnetica, si può pensare di complicare la configurazione precedente in modo da far lavorare il sensore ad effetto Hall nell'intorno di induzione nulla.

Nella fig. 1.4 si può vedere che sul nucleo toroidale trova posto un avvolgimento con n_2 spire pilotato dall'uscita dell'amplificatore (circuito secondario). All'ingresso dell'amplificatore vi è la tensione V_H generata per effetto Hall (molto piccola) dalla interazione tra la corrente di riferimento I_{REF} ed il campo magnetico impresso nel nucleo toroidale.

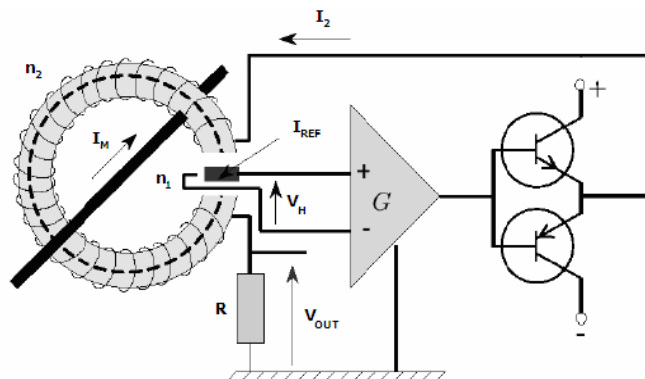


Fig. 1.14. Circuito magnetico per la sonda ad effetto Hall a compensazione di campo [Hal¹]

L'uscita dell'amplificatore pilota un generatore di corrente che imprime una corrente I_2 sull'avvolgimento secondario. L'avvolgimento è fatto in modo tale che la corrente I_2 imprime un campo magnetico che si oppone a quello generato dalla corrente incognita I_M . Se i due campi magnetici sono in equilibrio, vale a dire se $|I_M| n_1 = |I_2| n_2$, dove $n_1 = 1$, allora la tensione V_H è nulla.

In condizione di stabilità si ha quindi: $I_2 = I_M / n_2$, la corrente è n_2 volte più piccola di quella originaria e quindi più facilmente misurabile.

$$(1.10) \quad V_{OUT} = R \cdot (I_2)$$

Tra i vantaggi dell'uso di sonde ad effetto Hall, abbiamo: idoneo alla misura su inverter di potenza, molto lineare, dimensione anche molto ridotte, elevata velocità di risposta, costo contenuto, isolamento galvanico, ecc. L'unico difetto è che richiede un circuito di alimentazione

1.4.2.4.3. Encoder composto da sonde ad effetto Hall

L'encoder [DeD¹] per la misura della posizione angolare può essere fatto attraverso sonde ad effetto Hall. Un tipico esempio di un encoder di posizione angolare per una macchina a magneti permanenti è mostrata in fig. 1.15. Questo encoder di bassa risoluzione è normalmente utilizzato per fornire la commutazione per l'azionamento BLDC (**DC Brushless**). Tre sensori sono posizionati nella periferia dello statore, a distanza di 120° gradi elettrici uno dell'altro, in modo che un encoder con una risoluzione pari a 6 è ottenuto. I sensori di solito sono inseriti sullo statore con resina epossidica o montati su supporti.

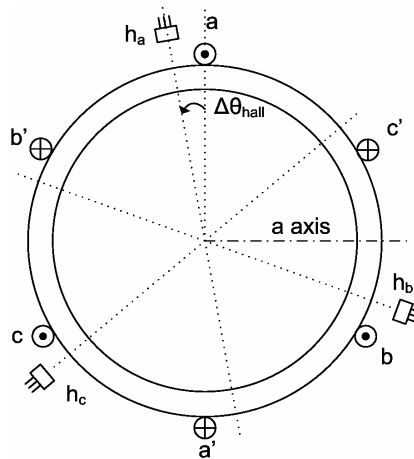


Fig. 1.15. Encoder per la misura di posizione angolare fatto da sonde ad effetto Hall [DeD¹]

In genere, l'encoder di posizione angolare sarà spostato in un angolo $\Delta\theta_{HALL}$ rispetto all'asse **a** di riferimento, quest'angolo può essere compensato attraverso un adeguato sistema di controllo.

L'accuratezza di detto encoder di posizione angolare dipende di vari fattori: il modo in cui i sensori sono montati, il tipo di macchina a magneti permanenti utilizzati, il numero di coppie di poli, il tipo di magnete, ecc. Per esempio, è molto difficile inserire i sensori su una macchina senza cave che su una macchina con cave poiché non ci sono i denti che aiutano come riferimento angolare. Un'altra difficoltà deriva dal fatto che sulle macchine ad alto numero di poli, l'angolo meccanico del quale i sensori devono essere spazati decrescono proporzionalmente alle coppie polari della macchina stessa.

Sebbene i sensori sono correttamente spazati non ci sono garanzie di accuratezza sulla misura: variazioni nel livello di flusso magnetico tra magneti adiacenti per effetti della temperatura o variazione dei livelli di isteresi tra i sensori peggiorano l'accuratezza dell'encoder. Inoltre, gli errori di misura dovuti alla distorsione del campo per effetto della reazione di armatura possono essere molto significativi.

1.5. Azionamenti con Motori Brushless

1.5.1. Generalità

In questa parte verranno presentate [Dis¹] le tecniche di controllo degli azionamenti riguardanti motori di tipo brushless; tali motori sono, essenzialmente, macchine sincrone nella cui struttura fisica non sono presenti le gabbie smorzatrici ed il cui circuito di eccitazione è inesistente in quanto sostituito da magneti permanenti. Nel seguito si introdurrà allora inizialmente il modello completo del settimo ordine della macchina sincrona sfruttando la Teoria di Park; successivamente si ricaverà il modello dei motori brushless particolarizzando (e semplificando) il modello generale.

Il campo di applicazione degli azionamenti brushless è circoscritto alle piccole potenze (tipicamente inferiori ai 50 kW): tale limitazione è data dalle dimensioni dei magneti permanenti che essi avrebbero per valori superiori di potenza ⁽¹⁾. Ciononostante, tali motori rivestono sempre più importanza in svariati campi applicativi, quali: macchine utensili a controllo numerico, automatismi industriali, robotica, ecc. Per tali applicazioni, l'uso dei motori brushless in luogo di quelli in corrente continua, che tradizionalmente venivano impiegati, consente alcuni vantaggi, fra i quali i più significativi sono:

- Velocità raggiungibili maggiori, a parità di potenza impiegata, di quelle dei motori in corrente continua in quanto non risultano essere limitate dalla presenza di collettori a spazzole;
- Peso, volume e sviluppo assiale ridotti rispetto ai motori in corrente continua; in tal modo sono presenti inerzie rotoriche minori e dunque, a parità di coppia generata, maggiori accelerazioni;
- Assenza virtuale di perdite nel rotore, il che si traduce in costanti termiche più basse; per tale motivo i motori brushless non necessitano di ventilazione forzata e possono dunque essere impiegati anche in ambienti speciali ⁽²⁾;
- Maggior semplicità costruttiva rispetto ai motori in corrente continua e miglior robustezza;
- Grazie all'assenza delle spazzole, necessitano manutenzioni molto meno frequenti rispetto ai motori in corrente continua.

L'unico svantaggio presentato risiede nel maggior costo dovuto ai magneti permanenti, il quale cresce al crescere della potenza.

¹ Ecco spiegato anche perché in tali motori non si pone alcuna gabbia smorzatrice.

² Ad esempio, in ambienti polverosi.

1.5.2 Modello della macchina sincrona

1.5.2.1 Struttura fisica ed equazioni descrittive

La macchina sincrona [Dis¹] è costituita da uno statore sul quale sono disposti degli avvolgimenti trifase simili a quelli della macchina asincrona e che generano un campo rotante; per non perdere di generalità si considererà un rotore anisotropo, ossia costituito da un certo numero di espansioni polari. I poli magnetici di rotore sono, per quanto riguarda la trattazione generale che qui si considera inizialmente, generati da un circuito di eccitazione in cui scorre una corrente i_F impressa dall'esterno. Esistono altri tipi di macchine sincrone, dette isotrope, in cui sul rotore sono disposti degli avvolgimenti simili a quelli di statore e che, percorsi dalla corrente i_F , generano un campo magnetico per il quale è possibile riconoscere formalmente una certa successione di poli. Nella presente trattazione si lavorerà sotto le seguenti ipotesi:

1. Si considererà una macchina anisotropa a $2n$ poli salienti;
2. Si supporrà la permeabilità magnetica dei tratti in ferro infinita;
3. Si considererà la macchina come sistema simmetrico ed equilibrato;
4. Si supporrà che la forza magnetomotrice di eccitazione, determinata dall'alternarsi dei poli di rotore al traferro della macchina, sia perfettamente sinusoidale;
5. Si supporrà di poter pensare gli avvolgimenti di statore come concentrati sui propri assi magnetici.

In Fig. 1.16 è rappresentata la struttura della macchina sincrona che ci si propone di analizzare. In particolare si nota come sullo statore sia presente un avvolgimento trifase simmetrico rappresentato da tre avvolgimenti monofase, uguali e posti a 120° l'uno rispetto all'altro; sul rotore è presente l'avvolgimento di eccitazione, percorso dalla corrente i_F , e la gabbia smorzatrice, studiata pensandola scomposta in due avvolgimenti equivalenti e mutuamente disaccoppiati, l'uno agente sull'asse diretto d e l'altro sull'asse in quadratura q .

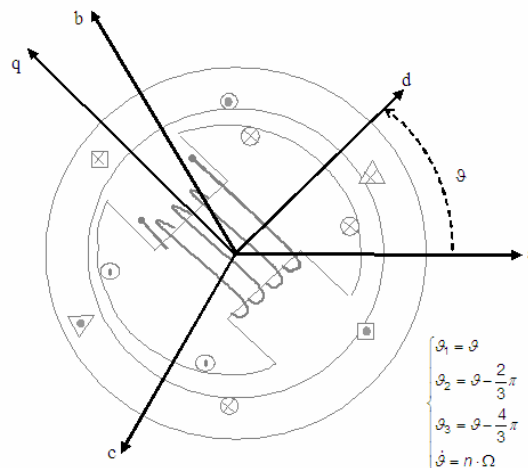


Fig. 1.16. Struttura della macchina sincrona anisotropa; la gabbia smorzatrice è schematizzata mediante due avvolgimenti, l'uno agente sull'asse d e l'altro sull'asse q e rappresentati circuitualmente mediante le coppie tensioni-correnti v_D, i_D e v_Q, i_Q ; gli angoli fra l'asse d e le varie fasi vengono definiti come indicato e considerati positivi in senso antiorario. [Dis¹]

Tenuto conto di tutte le resistenze, le autoinduttanze e le mutue induttanze, la Teoria mostra che le equazioni magneto-elettriche descrittive della macchina nel dominio delle fasi a,b,c risultano essere:

$$(1.11) \quad \begin{cases} [v] = [R][i] + p[\psi] \\ [\psi] = [L(\vartheta)][i] \end{cases}$$

In tali relazioni sono stati considerati a scopo di sintesi i seguenti vettori::

$$(1.12) \quad [v] = \begin{bmatrix} v_a \\ v_b \\ v_c \\ \frac{v_c}{v_f} \\ 0 \\ 0 \end{bmatrix}, \quad [i] = \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \\ i_D \\ i_Q \end{bmatrix}, \quad [\psi] = \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \\ \psi_f \\ \psi_D \\ \psi_Q \end{bmatrix}$$

Inoltre, in base alle ipotesi fatte, la matrice delle resistenze assume la forma:

$$(1.13) \quad [R] = \left[\begin{array}{ccc|ccc} R_s & 0 & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & R_f & 0 & 0 \\ 0 & 0 & 0 & 0 & R_D & 0 \\ 0 & 0 & 0 & 0 & 0 & R_Q \end{array} \right] = \begin{bmatrix} [R_{abc}] & [0] \\ [0] & [R_{fDQ}] \end{bmatrix}$$

e la matrice delle induttanze è:

$$(1.14) \quad [L(\vartheta)] = \left[\begin{array}{ccc|ccc} L_{aa}(\vartheta_1) & L_{ab}(\vartheta_3) & L_{ac}(\vartheta_2) & L_{af}(\vartheta_1) & L_{aD}(\vartheta_1) & L_{aQ}(\vartheta_1) \\ L_{ab}(\vartheta_3) & L_{bb}(\vartheta_2) & L_{bc}(\vartheta_1) & L_{bf}(\vartheta_2) & L_{bD}(\vartheta_2) & L_{bQ}(\vartheta_2) \\ L_{ac}(\vartheta_2) & L_{bc}(\vartheta_1) & L_{cc}(\vartheta_3) & L_{cf}(\vartheta_3) & L_{cD}(\vartheta_3) & L_{cQ}(\vartheta_3) \\ \hline L_{af}(\vartheta_1) & L_{bf}(\vartheta_2) & L_{cf}(\vartheta_3) & L_f & L_{fD} & 0 \\ L_{aD}(\vartheta_1) & L_{bD}(\vartheta_2) & L_{cD}(\vartheta_3) & L_{fD} & L_D & 0 \\ L_{aQ}(\vartheta_1) & L_{bQ}(\vartheta_2) & L_{cQ}(\vartheta_3) & 0 & 0 & L_Q \end{array} \right]$$

$$[L(\vartheta)] = \begin{bmatrix} [L_s(\vartheta)] & [M(\vartheta)] \\ [M(\vartheta)]_t & [L_r] \end{bmatrix}$$

Si noti come tutte le sottomatrici componenti la matrice delle induttanze, eccezion fatta per quella delle induttanze di rotore, siano dipendenti dall'ascissa angolare ϑ . Gli elementi di tali sottomatrici hanno espressione:

$$(1.15) \quad \begin{cases} L_{aa}(\vartheta_1) = I_o + L_{oo} + L \cos(2\vartheta_1) \\ L_{bb}(\vartheta_2) = I_o + L_{oo} + L \cos(2\vartheta_2) \\ L_{cc}(\vartheta_3) = I_o + L_{oo} + L \cos(2\vartheta_3) \end{cases} \quad (\text{Autoinduttanze di statore})$$

$$(1.16) \quad \begin{cases} L_{ab}(\vartheta_3) = -\frac{1}{2}L_{oo} + L \cos(2\vartheta_3) \\ L_{bc}(\vartheta_1) = -\frac{1}{2}L_{oo} + L \cos(2\vartheta_1) \\ L_{ca}(\vartheta_2) = -\frac{1}{2}L_{oo} + L \cos(2\vartheta_2) \end{cases} \quad (\text{mutue induttanze di statore})$$

$$(1.17) \quad \begin{cases} L_{af}(\vartheta_1) = L_{mf} \cos(\vartheta_1) \\ L_{bf}(\vartheta_2) = L_{mf} \cos(\vartheta_2) \\ L_{cf}(\vartheta_3) = L_{mf} \cos(\vartheta_3) \end{cases} \quad (\text{mutue induttanze statore-avv. di eccitaz.})$$

$$(1.18) \quad \begin{cases} L_{aD}(\vartheta_1) = L_{mD} \cos(\vartheta_1) \\ L_{bD}(\vartheta_2) = L_{mD} \cos(\vartheta_2) \\ L_{cD}(\vartheta_3) = L_{mD} \cos(\vartheta_3) \end{cases} \quad (\text{mutue induttanze statore-gabbia D})$$

$$(1.19) \quad \begin{cases} L_{aQ}(\vartheta_1) = -L_{mQ} \sin(\vartheta_1) \\ L_{bQ}(\vartheta_2) = -L_{mQ} \sin(\vartheta_2) \\ L_{cQ}(\vartheta_3) = -L_{mQ} \sin(\vartheta_3) \end{cases} \quad (\text{mutue induttanze statore-gabbia Q})$$

Nelle precedenti compare ovviamente l'induttanza associata al campo rotante (L_{oo}); per i termini non indicati ma che compaiono nelle (1.14) è da intendersi che essi sono costanti. Al sistema (1.11) vanno aggiunte le relazioni meccaniche:

$$(1.20) \quad \begin{cases} p\dot{\vartheta} = \frac{n}{j} \cdot (C - C_R) \\ p\vartheta = \dot{\vartheta} \end{cases}$$

Trattasi di un sistema differenziale dell'ottavo ordine, composto da 8 equazioni nelle otto incognite rappresentate dai sei flussi magnetici, dall'ascissa angolare ϑ e dalla velocità $\dot{\vartheta}$.

1.5.2.2 Trasformazione delle variabili

Verrà ora impiegata [Dis¹] la Trasformata di Park sugli assi rotanti per ottenere una macchina equivalente a collettore della macchina sincrona originaria; tenendo presente la particolare struttura di quest'ultima, si conclude che la trasformazione delle variabili secondo Park è da eseguirsi sulle sole variabili di statore.

Il sistema descrittivo (1.11) sul quale si va ad agire è riscrivibile come:

$$(1.21) \quad \begin{cases} [v_{abc}] = [R_{abc}][i_{abc}] + p[\psi_{abc}] \\ [v_{fDQ}] = [R_{fDQ}][i_{fDQ}] + p[\psi_{fDQ}] \\ [\psi_{abc}] = [L_s(\vartheta)][i_{abc}] + [M(\vartheta)][i_{fDQ}] \\ [\psi_{fDQ}] = [M(\vartheta)]_t[i_{abc}] + [L_r][i_{fDQ}] \end{cases}$$

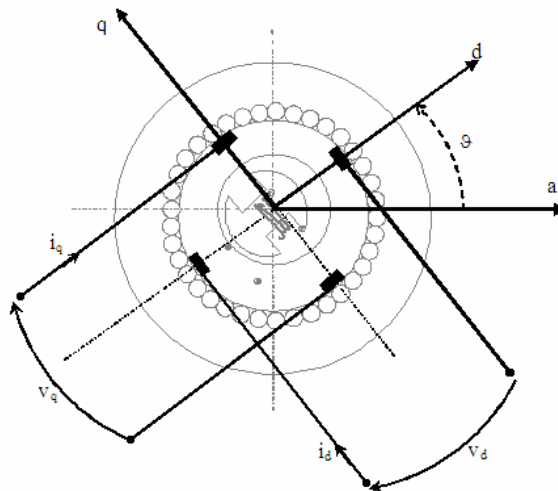


Fig.1.17. Macchina equivalente a collettore della macchina sincrona, ottenuta mediante la Trasformata di Park sugli assi rotanti [Dis¹]

La Trasformata di Park sugli assi rotanti è definita come:

(1.22)

$$F_{abc} = \begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix}; F_{qdo} = [K] \cdot \begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix} = K \cdot \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta - \frac{4}{3}\pi\right) \\ \sin(\theta) & \sin\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta - \frac{4}{3}\pi\right) \\ h & h & h \end{bmatrix} \cdot \begin{bmatrix} F_a \\ F_b \\ F_c \end{bmatrix}$$

Esistono 2 coppie di valori che vengono usati per K e h, con cui si può definire la trasformata di Park:

- 1-. Trasformazione invariante rispetto alla ampiezza $\Rightarrow K = \frac{2}{3}; h = \frac{1}{2}$
- 2-. Trasformazione invariante rispetto alla potenza $\Rightarrow K = \sqrt{\frac{2}{3}}; h = \frac{1}{\sqrt{2}}$

Applicando la Trasformata di Park sugli assi rotanti invariante rispetto alla ampiezza, dovuto ai tanti vantaggi rispetto a quella invariante in potenza (nell'invariante rispetto alla ampiezza si ha che la F_0 è pari alla componente omeopolare), si ottiene il sistema differenziale descrittivo seguente ⁽³⁾:

³ La matrice [B] ha espressione:

$$[B] = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$(1.23) \quad \begin{cases} [v_{dqo}] = [R_{abc}][i_{dqo}] + p[\psi_{dqo}] + \dot{\theta}[B][\psi_{dqo}] \\ [v_{fDQ}] = [R_{fDQ}][i_{fDQ}] + p[\psi_{fDQ}] \\ [\psi_{dqo}] = [L_s][i_{dqo}] + [M'] [i_{fDQ}] \\ [\psi_{fDQ}] = [M']_t [i_{dqo}] + [L_r][i_{fDQ}] \end{cases}$$

in esso risulta:

$$(1.24) \quad [L_s] = \begin{bmatrix} \frac{3}{2}(L_{oo} + L) + I_o & 0 & 0 \\ 0 & \frac{3}{2}(L_{oo} + L) + I_o & 0 \\ 0 & 0 & I_o \end{bmatrix} = \begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_o \end{bmatrix}$$

$$(1.25) \quad [M'] = \begin{bmatrix} \sqrt{\frac{3}{2}}L_{mf} & 0 & 0 \\ \sqrt{\frac{3}{2}}L_{mD} & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}}L_{mQ} & 0 \end{bmatrix} = \begin{bmatrix} M_f & 0 & 0 \\ M_D & 0 & 0 \\ 0 & M_Q & 0 \end{bmatrix}$$

Quindi la Trasformata di Park ha consentito di ottenere matrici delle induttanze tempo-invarianti (in esse non compare più la dipendenza da $\theta = \theta(t)$).

Il sistema (1.23) così ottenuto è descrittivo di una macchina equivalente a collettore (Fig. 1.17) in cui il rotore è identico a quello nel dominio originario (le variabili di rotore non sono state infatti trasformate), mentre lo statore è identico a quello di una macchina equivalente a collettore della macchina asincrona.

1.5.2.3 Bilancio energetico nel dominio di Park

Ricordando [Dis¹] che la matrice di Park, in quanto ortogonale, conserva le potenze, si vuole ora scrivere il bilancio energetico della macchina sincrona lavorando sulla macchina equivalente a collettore. Ci si aspetta un'espressione nella forma:

$$(1.26) \quad p(t) = p_J(t) + p_m(t) + \frac{dW}{dt}$$

dove W rappresenta l'energia elettromagnetica scambiata fra le varie parti interagenti, p_m è la potenza generata dalla macchina e p_J è la potenza dissipata per effetto Joule sotto forma di calore.

Sfruttando le equazioni precedentemente ricavate ⁽⁴⁾ si ha:

$$(1.27) \quad p(t) = v_d i_d + v_q i_q + v_f i_f + v_D i_D + v_Q i_Q$$

$$p(t) = \underbrace{R_s \cdot (i_d^2 + i_q^2) + R_f i_f^2 + R_D i_D^2 + R_Q i_Q^2}_{p_j} + \underbrace{i_d p \psi_d + i_q p \psi_q + i_f p \psi_f + i_D p \psi_D + i_Q p \psi_Q}_{\frac{dW}{dt}} + \underbrace{\dot{\psi}(i_q \psi_d - i_d \psi_q)}_{p_m}$$

L'espressione della coppia diviene allora:

$$(1.28) \quad C(t) = \frac{p_m}{\Omega} = p_m \cdot \frac{n}{g} = n \cdot (i_q \psi_d - i_d \psi_q)$$

1.5.3 Modello del motore brushless

Come già accennato al paragrafo 1.5.1, [Dis¹] i motori brushless presentano delle semplificazioni rispetto al modello generale del settimo ordine appena ricavato, e precisamente:

- Assenza delle gabbie smorzatrici; per tener conto di tale aspetto si dovrà imporre nel sistema descrittivo l'annullarsi delle correnti nelle gabbie equivalenti, ossia:

$$(1.29) \quad i_D = i_Q = 0$$

- Flusso di eccitazione costante, dovuto ai magneti permanenti;

$$(1.30) \quad \psi_f = \Psi_{MP} \Rightarrow p \psi_f = 0$$

Tale flusso risulta essere solidale con l'asse rotante d di Park, cioè:

$$(1.31) \quad \begin{cases} \overline{\psi}_f = \psi_{fd} \\ \psi_{fq} = 0 \end{cases}$$

A partire dal sistema descrittivo (1.23) nel dominio di Park, è allora possibile riscrivere le equazioni di statore come:

$$(1.32) \quad \begin{cases} v_{sd} = R_s \cdot i_{sd} + L_d p i_{sd} - \dot{\psi} \cdot L_q i_{sq} \\ v_{sq} = R_s \cdot i_{sq} + L_q p i_{sq} + \dot{\psi} \cdot (L_d \cdot i_{sd} + \Psi_{MP}) \\ \psi_{sd} = L_d \cdot i_{sd} + \Psi_{MP} \\ \psi_{sq} = L_q \cdot i_{sq} \end{cases}$$

⁴ Si osservi anche che per macchine aventi gli avvolgimenti di statore collegati a stella e con centro stella isolato (ciò che accade praticamente sempre) le correnti di statore sono a somma nulla, dunque la corrente omopolare è nulla, così come il flusso omopolare e la tensione omopolare. Si tralasceranno allora tutte queste componenti anche nel seguito, dunque il sistema descrittivo della macchina sincrona diviene del settimo ordine anziché dell'ottavo.

L'espressione (1.28) della coppia generata si particolarizza come:

$$(1.33) \quad \begin{aligned} C(t) &= n \cdot (i_q \psi_d - i_d \psi_q) = n \cdot (i_q \cdot i_d \cdot L_d + i_q \cdot \Psi_{MP} - i_d \cdot i_q \cdot L_q) \\ C(t) &= n \cdot [(L_d - L_q) \cdot i_d \cdot i_q + \Psi_{MP} \cdot i_q] \end{aligned}$$

Il termine proporzionale a $(L_d - L_q)$ è detto **coppia di anisotropia**, ed in effetti risulta nullo se la macchina considerata è isotropa; poiché in quest'ultimo caso si ha:

$$(1.34) \quad L_d = L_q = L_s$$

espressione nella quale L_s è detta *induttanza sincrona*, la (1.33) diviene:

$$(1.35) \quad C(t) = n \cdot \Psi_{MP} \cdot i_q$$

da cui si nota come la corrente statorica sull'asse d non concorre, in caso di isotropia della macchina, alla generazione della coppia; ciò significa che la corrente in quadratura è responsabile di tale coppia, mentre la corrente diretta ha il compito di fluire la macchina, esattamente come in un motore a corrente continua; questo conferma senz'altro l'importanza della visione, ottenuta mediante la Teoria di Park, della macchina sincrona mediante una macchina equivalente a collettore.

Grazie poi alla (1.34), è possibile esprimere il sistema descrittivo (1.32) compattandolo con l'ausilio dei fasori spaziali, ottenendo:

$$(1.36) \quad \begin{cases} \bar{v}_s = R_s \cdot \bar{i}_s + p \bar{\psi}_s + j \dot{\bar{\psi}}_s \\ \bar{\psi}_s = L_s \cdot \bar{i}_s + \Psi_{MP} \end{cases}$$

espressione valida, si ricorda, per i motori brushless isotropi.

Tornando alla macchina anisotropa (caso generale), si osserva che a regime risulta $\dot{\bar{\psi}} = \omega(\text{sincronismo})$ ed inoltre tutte le derivate temporali si annullano ($p=0$); allora, in tali condizioni, le (1.32) divengono:

$$(1.37) \quad \begin{cases} v_{sd} = R_s \cdot i_{sd} - \omega \cdot L_q i_{sq} \\ v_{sq} = R_s \cdot i_{sq} + \omega \cdot (L_d \cdot i_{sd} + \Psi_{MP}) \\ \psi_{sd} = L_d \cdot i_{sd} + \Psi_{MP} \\ \psi_{sq} = L_q \cdot i_{sq} \end{cases}$$

Definendo ora la forza elettromotrice indotta in condizioni di regime come:

$$(1.38) \quad E = \omega \cdot \Psi_{MP}$$

le equazioni della (1.37) sono infine riscrivibili come:

$$(1.39) \quad \begin{cases} v_{sd} = R_s \cdot i_{sd} - \omega \cdot L_q i_{sq} \\ v_{sq} = R_s \cdot i_{sq} + \omega \cdot L_d \cdot i_{sd} + E \\ \psi_{sd} = L_d \cdot i_{sd} + \Psi_{MP} \\ \psi_{sq} = L_q \cdot i_{sq} \end{cases}$$

Si osservi come la (1.38) permetta facilmente il calcolo del flusso che i magneti permanenti devono produrre; supponendo infatti che in condizioni di sincronismo e con motore a vuoto risulti $E \cong V_s$, dove V_s indica la tensione statorica applicata, si ha:

$$(1.40) \quad \Psi_{MP} = \frac{V_s}{\omega} = \frac{V_s}{2\pi f}$$

I motori brushless si suddividono in due categorie, denominate *AC-brushless* e *DC-brushless* a seconda della distribuzione degli avvolgimenti di statore e, di conseguenza, della forma d'onda assunta dal flusso concatenato di statore e dalle forze elettromotrici presenti in esso.

1.5.4. Azionamenti con motori "AC Brushless"

In questi tipi di motore brushless [Dis¹] gli avvolgimenti statorici risultano essere uniformemente distribuiti, dunque il flusso prodotto dai magneti permanenti di rotore e concatenato con essi presenta andamento sinusoidale per ciascuna fase statorica, ossia:

$$(1.41) \quad \begin{cases} \psi_{MP,a} = \Psi_{MP} \cdot \cos(\vartheta) \\ \psi_{MP,b} = \Psi_{MP} \cdot \cos\left(\vartheta + \frac{2}{3}\pi\right) \\ \psi_{MP,c} = \Psi_{MP} \cdot \cos\left(\vartheta - \frac{2}{3}\pi\right) \end{cases}$$

Grazie a tali andamenti, si intuisce come il controllo dell'azionamento di un AC-Brushless possa essere attuato vettorialmente, in analogia, per quanto detto al paragrafo precedente, con le tecniche viste per il motore in corrente continua; tale analogia è ovviamente significativa se si considera un motore di quest'ultimo tipo a magneti permanenti in quanto, così facendo, in entrambi l'eccitazione non risulta regolabile poiché il flusso prodotto è costante.

Si osservi che, perché l'analogia sia completa, è necessario che sia la sola corrente di statore sull'asse q a produrre la coppia; ciò impone l'adozione di un motore isotropo (come visto sopra) oppure l'imposizione di una corrente sull'asse diretto nulla: in tal caso la (1.33) porge:

$$(1.42) \quad C = n \cdot [(L_d - L_q) \cdot i_d \cdot i_q + \Psi_{MP} \cdot i_q] = \Psi_{MP} \cdot i_q = C_{\max}$$

esattamente come accade per il motore isotropo.

Tutte queste considerazioni suggeriscono come attuare il controllo dell'azionamento, come verrà ora mostrato; inizialmente si impone $i_{sd} = 0$ ed $i_{sq} = i_{max}$, ottenendo in tal modo che la coppia assuma il suo valor massimo, dato dalla (1.42). Tale situazione è rappresentabile, in base alle (1.39), con il diagramma vettoriale in Fig. 1.18.

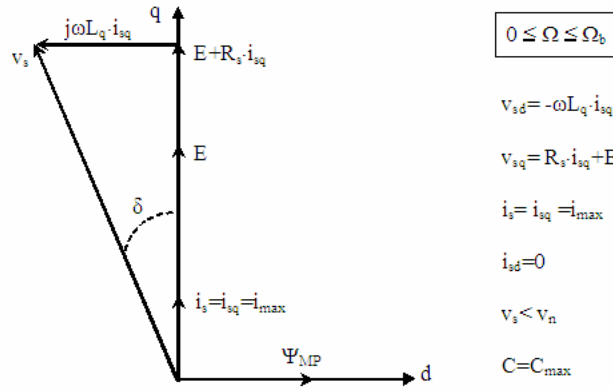


Fig. 1.18. Diagramma vettoriale relativo all'intervallo $0 \leq \Omega \leq \Omega_b$ [Dis¹]

All'aumentare della velocità meccanica Ω , aumentano di conseguenza, data la condizione di sincronismo, la quantità $\omega L_q \cdot i_{sq}$ e il modulo della tensione statorica; quando la velocità eguaglia il suo valore base, la tensione statorica raggiunge il suo valor massimo, pari al valore nominale ed imposto dal dispositivo statico di controllo. Se si vuole aumentare ancora la velocità di funzionamento, diviene necessario a questo punto variare il modulo della corrente statorica; in effetti non è più possibile mantenere la corrente i_q al valor massimo (e lavorare di conseguenza al valor massimo di coppia) in quanto un ulteriore aumento della forza elettromotrice E richiederebbe all'inverter una tensione superiore a quella massima da esso erogabile. E' consigliabile allora applicare una $i_d < 0$; in tal modo si ottiene:

$$(1.43) \quad i_s = \sqrt{i_{sd}^2 + i_{sq}^2} \leq i_{max}$$

E' così possibile aumentare la velocità mantenendo la tensione statorica pari al suo valor massimo e con una corrente statorica minore o al limite uguale al suo valor massimo, ma con una corrente in quadratura (dunque con una coppia) minore che nel caso si fosse imposto $i_d = 0$. Tale situazione è rappresentata dal diagramma vettoriale in Fig. 1.19.

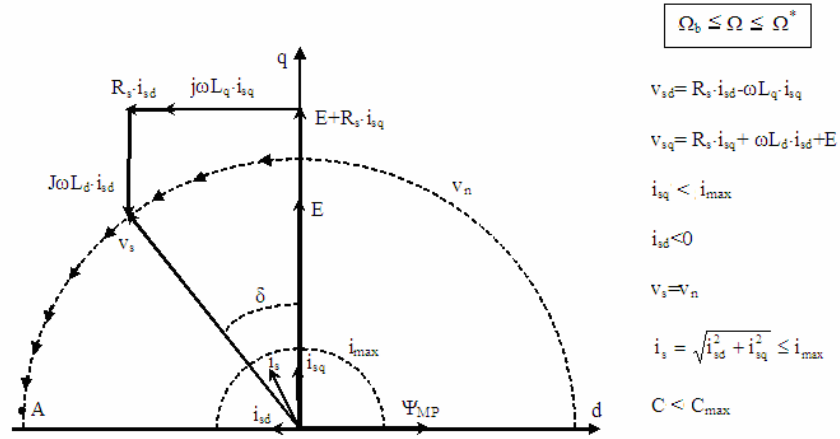


Fig. 1.19. Diagramma vettoriale relativo all'intervallo $\Omega_b \leq \Omega \leq \Omega^*$ [Dis¹]

Operativamente, quanto appena esposto significa spostare la punta del vettore $\bar{V}_s$ fino al punto A (caratterizzato da una velocità Ω^*) il più velocemente possibile; in base a tale vincolo, si assegna al regolatore di velocità, che controlla l'inverter utilizzato, la legge di controllo della corrente i_d e si ricava la i_q tale che, dalla (1.43):

$$(1.44) \quad i_q = \sqrt{i_{max}^2 - i_d^2}$$

In tale campo di velocità si parla, impropriamente, di deflussaggio, in analogia con quanto avviene per il controllo del motore in corrente continua; in effetti qui il flusso non può essere regolato, ma tutto avviene come se lo fosse, anche se in realtà a diminuire è la corrente i_q .

Quando la velocità ha raggiunto il valore Ω^* , risulterà:

$$(1.45) \quad \omega^* L_d \cdot i_{sd} = E^* = \omega^* \cdot \Psi_{MP} \Rightarrow i_{sd} = i_{sd}^* = \frac{\Psi_{MP}}{L_{sd}} = \text{costante}$$

Una volta superato tale valore, corrispondente come detto al punto A in cui $\delta \cong \pi/2$, è possibile aumentare ulteriormente la velocità diminuendo ancora la corrente i_d finché risulta $i_d = -i_{max}$ cioè finché la velocità meccanica eguaglia il suo valore massimo. In quest'ultimo campo di regolazione si ha, trascurando la caduta di tensione su R_s :

$$(1.46) \quad i_q \cong \frac{V_{max}}{\omega L_q}$$

cioè la corrente in quadratura ha andamento proporzionale ad $1/\omega$, ossia ad $1/\Omega$.

Date tutte le precedenti considerazioni, il campo di operatività dell'azionamento del motore AC-brushless è quello rappresentato in Fig. 1.20; nella realtà, in effetti, la presenza dei magneti permanenti, la cui permeabilità magnetica è molto prossima, per costituzione costruttiva, a quella dell'aria fa sì che il traferro esistente effettivamente fra lo statore ed il rotore sia ampio e che dunque le

induttanze in gioco siano modeste; essendo inoltre anche le resistenze degli avvolgimenti alquanto ridotte, ne risulta un'impedenza complessiva di valore non molto elevato. Di conseguenza la velocità meccanica massima risulta poco superiore al valore nominale, dunque la possibilità di estendere il tratto a potenza costante oltre Ω_n non è ripagata dall'aumento di complessità del regolatore; si preferisce dunque, in genere, considerare un campo di regolazione della velocità da 0 al valore Ω_n .

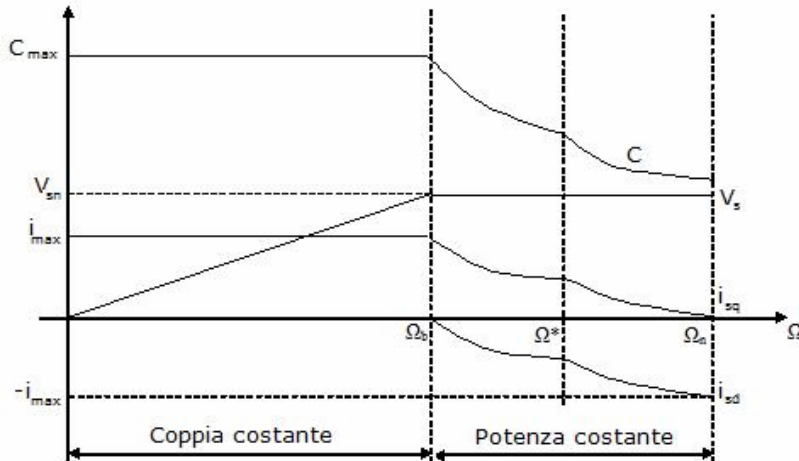


Fig. 1.20. Campo di operatività dell'azionamento con motore AC-Brushless [Dis¹]

L'architettura di base per il controllo appena descritto comprenderà dunque un inverter in grado di fornire le corrette correnti di fase al motore in base ai segnali di comando che giungono ai suoi switch e sono imposti dal regolatore di velocità; si noti come sia necessario conoscere istante per istante la posizione angolare del rotore dell'AC-Brushless, in quanto tale angolo serve al blocco " $\underline{K}$ " e " $\underline{K}^{-1}$ " per trasformare ed antitrasformare rispettivamente, i segnali di comando degli switch. Lo schema a blocchi relativo a tale controllo è riportato in Fig. 1.21.

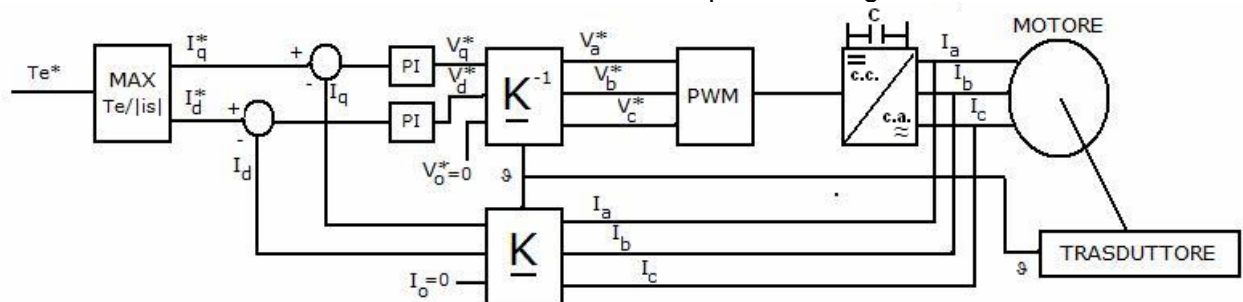


Fig. 1.21. Schema a blocchi del controllo del motore AC-Brushless

1.5.5. Azionamenti con motori "DC Brushless"

In tali motori, [Dis¹] aventi struttura isotropa, gli avvolgimenti di statore sono di tipo concentrato (solitamente si impiegano soluzioni costruttive che prevedono una cava per polo e per fase), dunque il flusso prodotto dai magneti permanenti e concatenato con essi risulta idealmente ad andamento triangolare e le forze elettromotrici in ciascuna fase presentano un andamento rettangolare ⁽⁵⁾. Nella realtà, in effetti, gli andamenti dei flussi concatenati non presentano andamenti perfettamente triangolari ⁽⁶⁾, dunque le forze elettromotrici risultano essere trapezoidali, da cui il nome alternativo di "Brushless trapezio". In Fig.1.22 sono riportati sia gli andamenti reali che quelli ideali.

Le forze elettromotrici sono esprimibili come:

$$(1.47) \quad \begin{cases} e_a = \Omega \cdot \Psi_{MP} \cdot f(\vartheta) \\ e_b = \Omega \cdot \Psi_{MP} \cdot f\left(\vartheta - \frac{2}{3}\pi\right) \\ e_c = \Omega \cdot \Psi_{MP} \cdot f\left(\vartheta + \frac{2}{3}\pi\right) \end{cases}$$

con f opportuna funzione della variabile angolare ϑ , misurata a partire dall'asse magnetico dei poli di rotore e positiva in senso antiorario.

Indicando con $\Psi_{MP,a}$, $\Psi_{MP,b}$, $\Psi_{MP,c}$ i flussi concatenati con gli avvolgimenti di statore, la coppia generata assume l'espressione:

$$(1.48) \quad \begin{aligned} C &= i_a \cdot p \Psi_{MP,a} + i_b \cdot p \Psi_{MP,b} + i_c \cdot p \Psi_{MP,c} \\ C &= n \cdot \left\{ i_a \cdot \Psi_{MP,a} \cdot f(\vartheta) + i_b \cdot \Psi_{MP,b} \cdot f\left(\vartheta + \frac{2}{3}\pi\right) + i_c \cdot \Psi_{MP,c} \cdot f\left(\vartheta - \frac{2}{3}\pi\right) \right\} \end{aligned}$$

⁵ Infatti esse sono ottenute per derivazione dal flusso.

⁶ Ciò presupporrebbe il dover considerare gli avvolgimenti statorici come puntiformi.

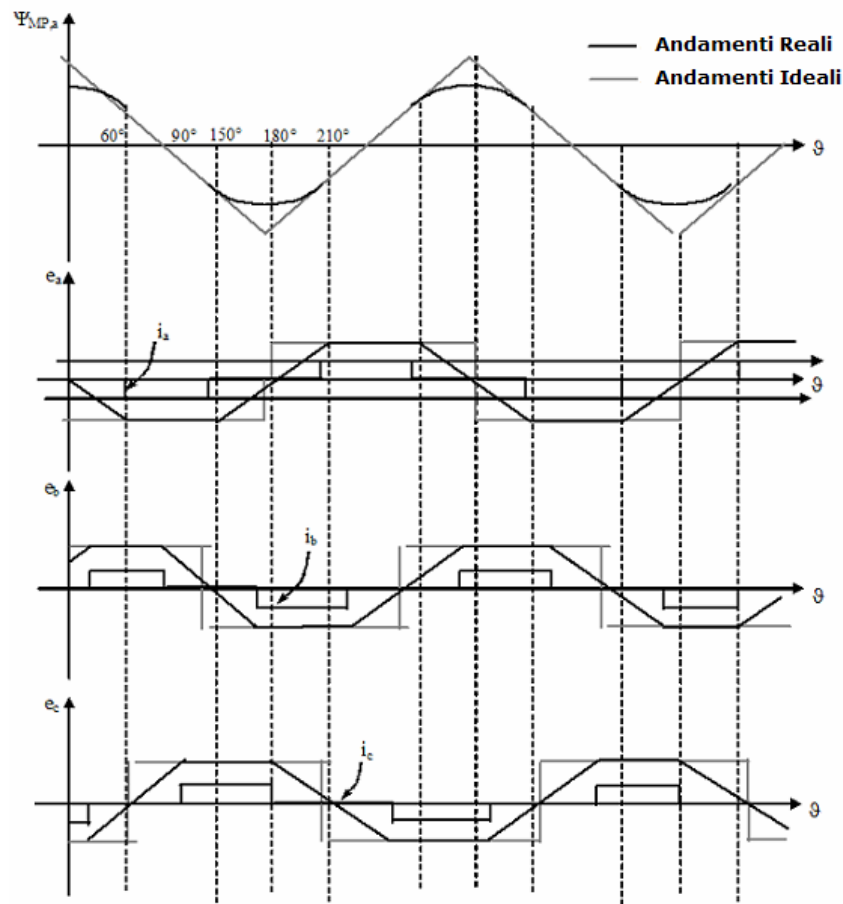


Fig. 1.22. Andamenti reali ed ideali del flusso prodotto dai magneti permanenti e concatenato con l'avvolgimento della fase a di statore e andamenti delle forze elettromotrici, ideali e reali, in tutte e tre le fasi di statore e delle relative correnti in un motore DC-Brushless [Dis¹]

Controllando le tre correnti statoriche in modo che abbiano lo stesso segno delle corrispondenti forze elettromotrici quando queste sono costanti e che siano nulle quando hanno pendenza diversa da zero come indicato nella stessa Fig.1.22, si riesce ad ottenere una coppia costante; questo implica che il regolatore impiegato per il controllo debba imporre all'inverter una commutazione ogni 60° .

Per realizzare tale tipo di controllo si impiega un inverter di tipo CPRPWM per le basse potenze ed un inverter CSI per le potenze più significative ⁽⁷⁾; dagli andamenti di Fig.1.22 si nota come, in ogni intervallo di commutazione, siano non nulle solamente due correnti di fase: il circuito equivalente dell'insieme inverter+motore è rappresentato in Fig.1.23 (in tale figura l'inverter è rappresentato con struttura ideale).

⁷ Ma, ovviamente, che giustificano l'adozione di un motore brushless.

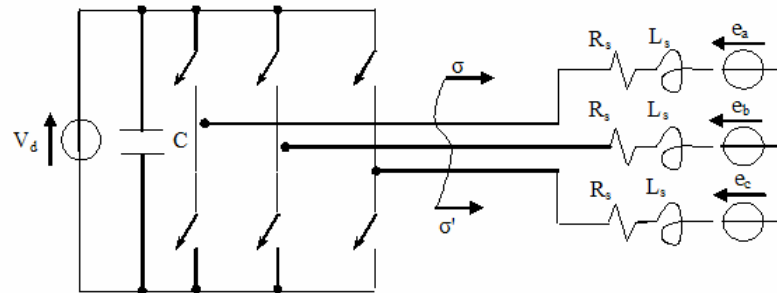


Fig. 1.23. Circuito equivalente del sistema inverter+motore DC-Brushless [Dis¹]

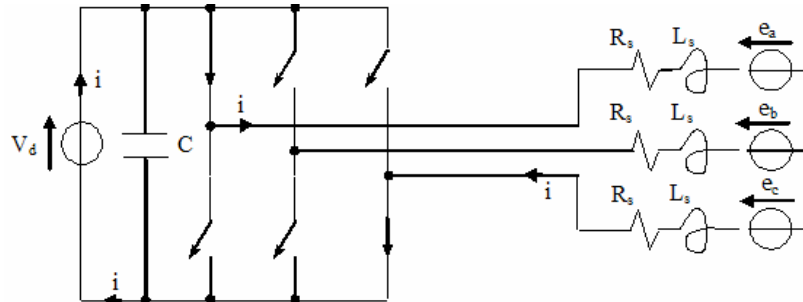


Fig. 1.24. Circuito equivalente del sistema inverter+motore DC-Brushless durante la fase di salita della corrente [Dis¹]

La sequenza delle commutazioni degli switch dell'inverter è eseguita secondo la seguente procedura:

1. In base all'angolo ϑ , un dispositivo detto *commutatore* riconosce in quale delle fasi la forza elettromotrice è massima negativa e considera tale fase come "fase di riferimento" da paragonare al riferimento di corrente esterno;
2. La corrente nella "fase di riferimento" viene mantenuta costante aprendo e chiudendo alternativamente gli switch della fase che presenta forza elettromotrice concatenata fra sé stessa e la "fase di riferimento" massima positiva;
3. In posizione mediana all'interno dell'intervallo di 120° considerato si ha la commutazione (dunque vi è una commutazione ogni 60°).

I circuiti equivalenti durante le fasi di salita della corrente, discesa della corrente e commutazione sono riportati in Fig. 1.24, 1.25 e 1.26; a titolo d'esempio, si sono riportati tali circuiti considerando la fase c come "fase di riferimento" e realizzando la commutazione dalla fase a alla fase b.

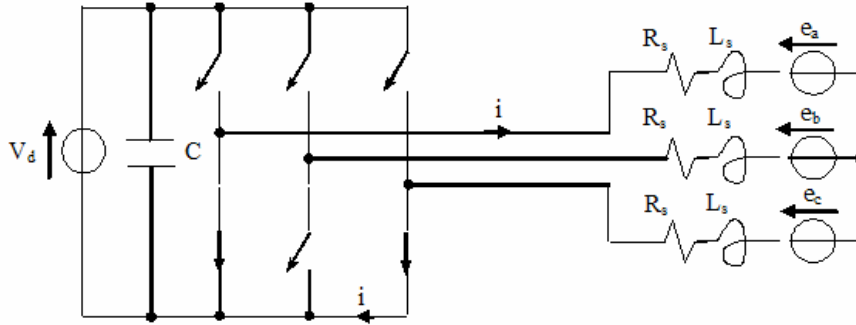


Fig. 1.25. Circuito equivalente del sistema inverter+motore DC-Brushless durante la fase di discesa della corrente [Dis¹]

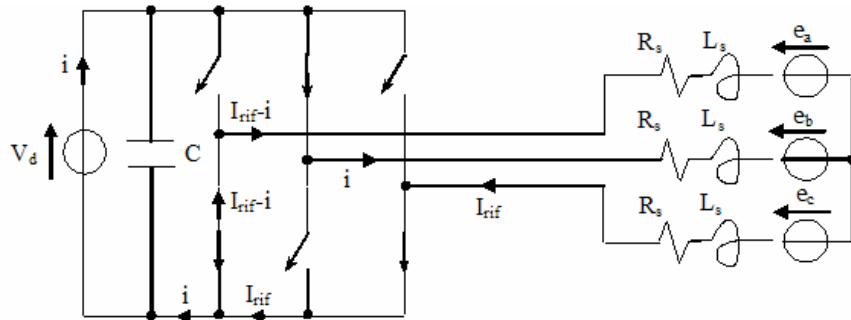


Fig. 1.26. Circuito equivalente del sistema inverter+motore DC-Brushless durante la commutazione fra la fase a e la fase b [Dis¹]

Per la maglia insistente durante la fase di salita della corrente (Fig. 1.24) si ha:

$$(1.49) \quad \begin{aligned} V_d &= R_s \cdot i + L_s p i + e_a + R_s \cdot i + L_s p i - e_c = 2 \cdot (R_s \cdot i + L_s p i) + (e_a - e_c) \\ V_d &= 2 \cdot (R_s \cdot i + L_s p i) + E \end{aligned}$$

Mentre per la maglia insistente durante la fase di discesa della corrente (Fig. 1.25) si ha:

$$(1.50) \quad \begin{aligned} 0 &= R_s \cdot i + L_s p i - e_a + R_s \cdot i + L_s p i + e_c = 2 \cdot (R_s \cdot i + L_s p i) + (e_c - e_a) \\ 0 &= 2 \cdot (R_s \cdot i + L_s p i) - E \end{aligned}$$

Questo comporta che dalla sezione σ - σ' il sistema viene visto come se fosse un motore in corrente continua. In effetti, mediante tale controllo, l'espressione (1.48) della coppia diviene:

$$(1.51) \quad C = 2n \cdot i \cdot \Psi_{MP}$$

simile all'espressione della coppia fornita da un motore in corrente continua.

Lo schema a blocchi del controllo e dell'azionamento è rappresentato in Fig.1.27; in esso compare, come anticipato poco sopra, un opportuno commutatore, ossia un dispositivo che, in base al sestante in cui si sta operando (rilevato mediante misura dell'angolo ϑ), "decide" come commutare gli switch dell'inverter, a partire dal valore di al duty-cycle δ imposto dal regolatore di corrente.

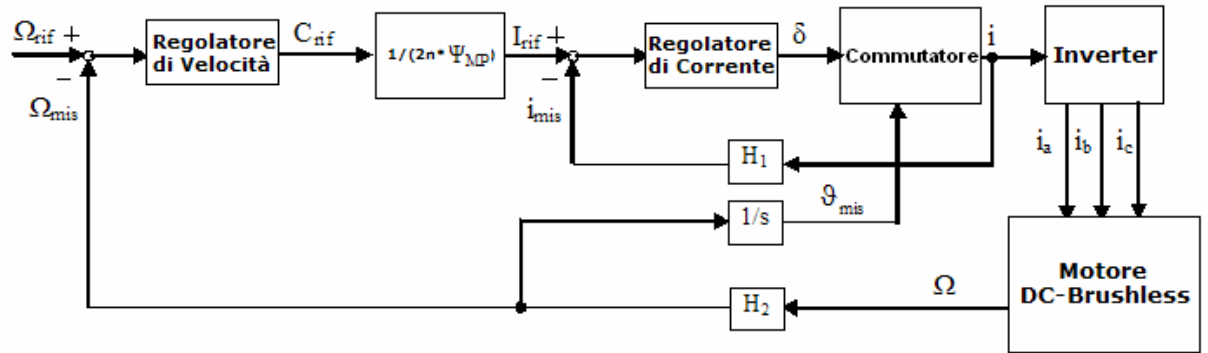


Fig. 1.27. Schema a blocchi dell'azionamento con motore DC-Brushless; δ indica il duty-cycle [Dis¹]

Si noti come il controllo impiegato è di tipo scalare, a differenza di quello impiegato per l'AC-brushless; in effetti qui non è impiegabile il controllo vettoriale in quanto le grandezze elettromagnetiche in gioco non presentano, come visto, andamento sinusoidale.

L'azionamento così rappresentato presenta un problema fondamentale: le commutazioni (maglia rappresentata in Fig. 1.28) impongono dei transitori RL governati da forzanti del tipo:

$$(1.52) \quad V_d - (e_j - e_k) = V_d - \Omega \cdot \Psi_{MP} \cdot [f(\vartheta_j) - f(\vartheta_k)]$$

relazione nella quale, ovviamente, i pedici j e k indicano le fasi a,b,c. Al variare della velocità meccanica variano dunque anche tali forzanti; in particolare, all'aumentare della velocità aumenta la differenza $e_j - e_k$, dunque le forzanti si riducono e, al limite, per $e_j - e_k = V_d$ (massima tensione disponibile) la corrente i si annullerebbe. La variabilità delle forzanti durante i transitori di commutazione genera dei *buchi di coppia* (cfr. fig. 1.28); addirittura, per $i < 0$, si otterrebbe, durante tali transitori, una coppia frenante che ridurrebbe il valore della coppia generata. Per evitare problemi si impone allora una velocità Ω_{max} tale che risulti $e_j - e_k \leq V_d$; inoltre, per ridurre al minimo i buchi di coppia, si può imporre che la commutazione avvenga in anticipo rispetto agli istanti temporali in cui le forze elettromotrici delle fasi di riferimento abbandonano il tratto in cui sono costanti.

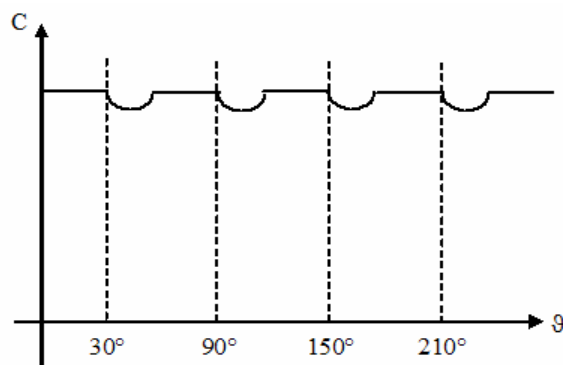


Fig. 1.28. Buchi di coppia dovuti alle commutazioni [Dis¹]

1.6. Conclusioni

I **Motori “AC Brushless”** sono diventati molto più maturi nell’aspetto tecnologico, dovuto al fatto che questi motori sono privi del ripple di coppia che si produce nei DC Brushless. Contrariamente, la necessità di misurare la posizione angolare ad alta risoluzione richiede l’adozione di costosi encoder ad alta risoluzione montati sull’albero del motore, il quale fa aumentare il costo di installazione e manutenzione dell’azionamento. Lo scopo di questa tesi è l’eliminazione di questi costosi encoder ad alta risoluzione; per questo motivo si è adottato una sistemazione economica a bassa risoluzione attraverso sonde ad effetto Hall.

L’adozione di una sistemazione economica a bassa risoluzione permette attraverso l’encoder composto da 2 sonde ad effetto Hall la misura della posizione angolare a velocità piena e carico di funzionamento, anche se questa soluzione non può essere utilizzata per applicazioni di posizionamento. Gli algoritmi per la stima della posizione e velocità angolare che vengono utilizzati insieme a questi encoder, possono essere classificati in due gruppi: il primo basato sull’extrapolazione ed il secondo basato sul rilevamento della forza elettromotrice. Il primo gruppo è caratterizzato da un ritardo nella stima della velocità, mentre il secondo presenta una dipendenza dai parametri elettrici ed ridotte prestazione alle basse velocità, poiché in questa regione la forza elettromotrice non è facilmente rilevabile. In riferimento a questo, sarebbe un significativo risultato determinare un algoritmo il quale possa stimare la velocità angolare con un ridotto ritardo, che garantisca soddisfacenti prestazioni alle basse velocità e non abbia dipendenza dei parametri elettrici.

L’algoritmo proposto per la stima della posizione e velocità angolare attraverso un encoder composto da 2 sonde binarie ad effetto Hall sarà analizzato nel capitolo 2 e costituisce il principale contributo di questa tesi.

Capitolo 2

Procedimenti per la stima di velocità e posizione

2.1. Introduzione

In questo capitolo si descrive il metodo e l'algoritmo utilizzato per la stima della posizione e velocità angolare. L'elaborazione del segnale che permette di ottenere l'informazione dal trasduttore (encoder composto da 2 sonde ad effetto Hall) sono presentati nel paragrafo 2.2; nel paragrafo 2.3 si introduce e si descrive il Vector Tracking Observer VTO (Osservatore ad inseguimento vettoriale), il quale viene usato per ottenere la stima della posizione e velocità angolare; il paragrafo 2.4 presenta le problematiche che emergono dall'applicazione del VTO per la misura della posizione e velocità angolare, come sono: problema di quantizzazione, gain scheduling e disaccoppiamento delle armoniche di quantizzazione. Per ultimo, nel paragrafo 2.5 si presenta le conclusioni sulle prestazioni del VTO nella misura della posizione e velocità angolare.

2.2. Elaborazione del segnale attraverso 2 sonde ad effetto Hall

Se abbiamo un encoder costituito da 2 sonde ad effetto Hall sfasate tra di loro 90° elettrici, questo permette una risoluzione di 90° elettrici e di individuare quattro settori entro i 360° elettrici. Nella figura 2.1 si mostra il rapporto che esiste tra la posizione angolare e gli stati delle sonde ad effetto Hall. Ogni settore può essere riconosciuto attraverso due bits (H_1, H_2) di informazione, che rappresentano la situazione di entrambe i sensori (1 significa che il sensore è acceso e 0 significa che è spento).

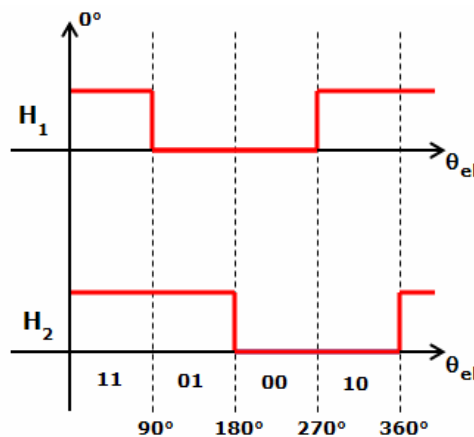


Fig. 2.1. Stati delle sonde ad effetto Hall entro un periodo elettrico

L'informazione ottenuta dalle sonde ad effetto Hall (parola composta da due bits) definisce quattro possibili posizioni che un vettore Hall rotante quantizzato può occupare in un sistema di riferimento stazionario detto " $\alpha\beta$ ".

Per trasformare gli stati delle due sonde ad effetto Hall (H_1 e H_2), nel sistema di riferimento stazionario " $\alpha\beta$ " è opportuno rendere simmetrica la forma d'onda ottenuta dalle sonde rispetto a zero, in modo che il centro del vettore Hall coincida con l'origine del sistema di riferimento. A questo scopo è necessario realizzare la seguente operazione:

$$(2.1) \quad \begin{pmatrix} H_\alpha \\ H_\beta \end{pmatrix} = 2 \cdot \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

dove H_α e H_β , sono direttamente le forme d'onda trasformate al sistema di riferimento stazionario " $\alpha\beta$ ", le quale sono pari a 1 quando il relativo sensore è acceso (on-state) e pari a -1 quando il sensore è spento (off-state). Poiché vi sono quattro combinazioni possibili di H_1 e H_2 , ci sono anche quattro possibili combinazioni per H_α e H_β . La figura 2.2 mostra un diagramma delle forme d'onda di H_α e H_β entro un periodo elettrico.

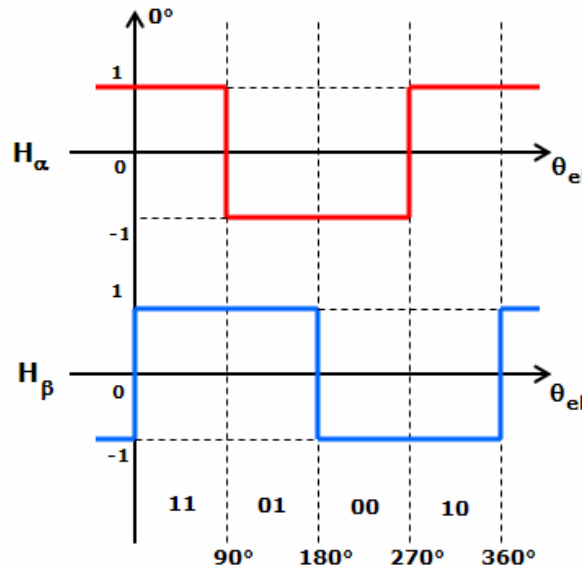


Fig. 2.2. Diagramma delle forme d'onda di H_α e H_β entro un periodo elettrico

Se H_α e H_β sono tracciati nel sistema di riferimento stazionario " $\alpha\beta$ ", si può ottenere il luogo del vettore spaziale Hall, il quale può essere definito come:

$$(2.3) \quad \vec{H}_{\alpha\beta} = H_\alpha + j H_\beta$$

Nella figura 2.3 si può vedere la posizione del vettore Hall ($\vec{H}_{\alpha\beta}$), che occupa la posizione particolare ai 45° elettrici rispetto all'asse di riferimento α , questo indica

che l'asse magnetico è da qualche parte all'interno del settore individuato dalla parola digitale 00. La posizione è interamente identificata dai due bits e l'associazione di parole digitali e vettori in fig. 2.3 è consistente con le figure 2.1 e 2.2.

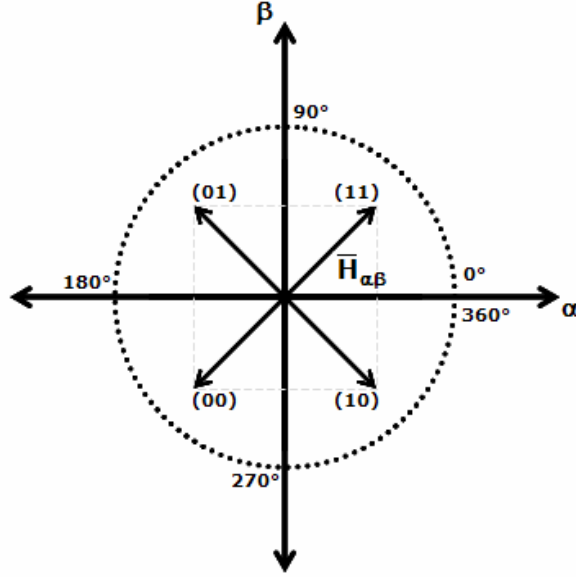


Fig. 2.3. Posizione del vettore Hall ($\vec{H}_{\alpha\beta}$) nel sistema di riferimento stazionario “αβ”

Un importante aspetto da vedere quando si confrontano le figura 2.1 e 2.3 è quello di ogni vettore Hall si dirige verso il centro del settore associato. Questo sembra di essere in apparente contrasto con l'idea che la informazione della posizione esatta si ottiene solo allo stato di transizione dei sensori, e non al punto medio di ogni settore. Una analisi delle armoniche spaziale del vettore rotante quantizzato chiarisce questo punto. Il modello delle armoniche è stato elaborato in [Tes¹] per un vettore rotante quantizzato con un generico numero intero N di vettori quantizzati in un giro (in questo caso particolare N = 4). Siccome $\vec{H}_{\alpha\beta}$ è una funzione continua e spazialmente periodica, si può rappresentare in termini di una Serie di Fourier Spaziale come:

$$(2.4) \quad \vec{H}_{\alpha\beta}(\theta_{el}) = \sum_{k=-\infty}^{+\infty} \vec{H}_{\alpha\beta}^k(\theta_{el}) = \sum_{k=-\infty}^{+\infty} a_k e^{jk\theta_{el}}$$

dove $\vec{H}_{\alpha\beta}^k(\theta_{el})$ è la k-esima armonica del vettore Hall. I coefficienti a_k sono definiti da seguito:

$$(2.5) \quad a_k = \frac{1}{2\pi} \int_0^{2\pi} \vec{H}_{\alpha\beta}(\theta_{el}) e^{-jk\theta_{el}} d\theta_{el}$$

Dato che $\vec{H}_{\alpha\beta}$ cambia il corso ogni 90° elettrici, l'integrale dell'eq. 2.5 può essere sostituito per una sommatoria di integrali su ogni settore di 90° elettrici. Così, a_k diventa:

$$(2.6) \quad a_k = \frac{1}{2\pi} \sum_{n=1}^4 \vec{H}_{\alpha\beta}(n) \int_{2\pi \frac{n}{4}}^{2\pi \frac{n+1}{4}} e^{-jk\theta_{el}} d\theta_{el}$$

Questa espressione può essere semplificata analizzando l'integrale e riconoscendo che $\vec{H}_{\alpha\beta} = e^{j\left(2\pi \frac{n}{4}\right)}$:

$$(2.7) \quad a_k = \frac{1}{2\pi k} \sum_{n=1}^4 e^{j2\pi \left(\frac{n}{4}\right)} \cdot \left(e^{-jk2\pi \left(\frac{n+1}{4}\right)} - e^{-jk2\pi \left(\frac{n}{4}\right)} \right)$$

I valori di a_k possono essere calcolati per valori interi di K:

$$(2.8) \quad a_k = \begin{cases} 0 & k \neq 4m+1 \\ \frac{1}{k} e^{-j\frac{\pi}{4}} & k = 4m+1 \end{cases} \Rightarrow m \in \{0; \pm 1; \pm 2; \dots\}$$

Finalmente, si ha la espressione totale per la Serie di Fourier Spaziale di un vettore Hall rotante quantizzato:

$$(2.9) \quad \begin{aligned} \vec{H}_{\alpha\beta}(\theta_{el}) = & e^{j\left(\theta_{el} - \frac{\pi}{4}\right)} - \frac{1}{3} e^{-j\left(3\theta_{el} + \frac{\pi}{4}\right)} + \frac{1}{5} e^{j\left(5\theta_{el} - \frac{\pi}{4}\right)} - \frac{1}{7} e^{-j\left(7\theta_{el} + \frac{\pi}{4}\right)} + \frac{1}{9} e^{j\left(9\theta_{el} - \frac{\pi}{4}\right)} - \dots \\ & \dots - \frac{1}{(4k-1)} e^{-j\left((4k-1)\theta_{el} + \frac{\pi}{4}\right)} + \frac{1}{(4k+1)} e^{j\left((4k+1)\theta_{el} - \frac{\pi}{4}\right)} \end{aligned}$$

L'equazione 2.9 spiega l'apparente contrasto tra le figure 2.1 e 2.3; la componente fondamentale $\vec{H}_{\alpha\beta}^1(\theta_{el})$ è in ritardo 45° elettrici rispetto al vettore rotante quantizzato $\vec{H}_{\alpha\beta}$, e poi è possibile vedere che le armoniche con segno negativo

$\left(-\frac{1}{(4k-1)} e^{-j\left((4k-1)\theta_{el} + \frac{\pi}{4}\right)} \right)$ hanno un verso controrotante rispetto alla armonica

fondamentale; per esempio la quinta armonica (negativa) fa 5 giri in senso controrotante nello stesso tempo che la fondamentale fa 1 giro. Se la componente fondamentale è rintracciata senza nessun ritardo, è pertanto corretto che il vettore Hall sia collocato come in figura 2.3.

Se per esempio abbiamo una componente fondamentale col ritardo sopra detto, con un angolo $\theta_{el}=60^\circ$ elettrici si può ottenere il vettore Hall rotante quantizzato descritto :

Tab. 2.1. Serie di Fourier Spaziale di un vettore Hall rotante quantizzato, per un $\theta_{el}=60^\circ$

Armonica	$\vec{H}_{\alpha\beta}(\theta_{el} = 60^\circ)$	(H_α, H_β)
Fondamentale	$e^{j\left(\theta_{el} - \frac{\pi}{4}\right)}$	(0.96593,0.25882)
3	$-\frac{1}{3}e^{-j\left(3\theta_{el} + \frac{\pi}{4}\right)}$	(0.23570,0.23570)
5	$\frac{1}{5}e^{j\left(5\theta_{el} - \frac{\pi}{4}\right)}$	(-0.05176,-0.19319)
7	$-\frac{1}{7}e^{-j\left(7\theta_{el} + \frac{\pi}{4}\right)}$	(0.03697,0.13799)
9	$\frac{1}{9}e^{j\left(9\theta_{el} - \frac{\pi}{4}\right)}$	(-0.07857,0.07857)
11	$-\frac{1}{11}e^{-j\left(11\theta_{el} + \frac{\pi}{4}\right)}$	(-0.08781,-0.02353)
13	$\frac{1}{13}e^{j\left(13\theta_{el} - \frac{\pi}{4}\right)}$	(0.07430,0.01991)
15	$-\frac{1}{15}e^{-j\left(15\theta_{el} + \frac{\pi}{4}\right)}$	(0.04714,0.04714)
17	$\frac{1}{17}e^{j\left(17\theta_{el} - \frac{\pi}{4}\right)}$	(-0.01522,-0.05682)
19	$-\frac{1}{19}e^{-j\left(19\theta_{el} + \frac{\pi}{4}\right)}$	(0.01362,0.05084)
21	$\frac{1}{21}e^{j\left(21\theta_{el} - \frac{\pi}{4}\right)}$	(-0.03367,0.03367)
23	$-\frac{1}{23}e^{-j\left(23\theta_{el} + \frac{\pi}{4}\right)}$	(-0.04199,-0.01125)
25	$\frac{1}{25}e^{j\left(25\theta_{el} - \frac{\pi}{4}\right)}$	(0.03864,0.01035)

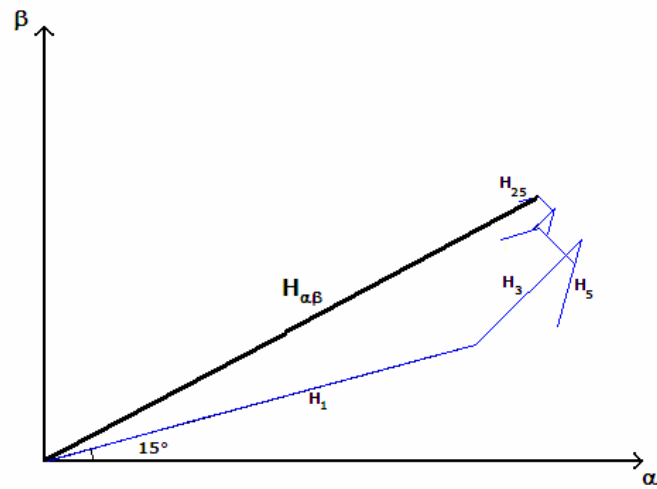


Fig. 2.4. Vettore Hall rotante quantizzato, partendo dalla sua componente fondamentale fino alla 25^{ma} armonica.

Con la finalità di rintracciare questa componente fondamentale $\vec{H}_{\alpha\beta}^1(\theta_{el})$, è utilizzato un Vector Tracking Observer VTO (Osservatore ad inseguimento vettoriale).

2.3. Vector-Tracking Observer VTO

L'osservatore ad inseguimento vettoriale (Vector-Tracking Observer, VTO) è un adattamento non-lineare dell'osservatore asintotico di Luenberger, in cui per rilevare la fase è usato un prodotto vettoriale, simile a quello utilizzato per il RDC (Resolver to Digital Converter) nel paragrafo 1.4.2.3. L'osservatore ad inseguimento vettoriale è stato ampiamente utilizzato nel settore del controllo di posizione sensorless per macchine AC, [Jan¹], [Deg¹] e [Kim¹]. Nel prossimo paragrafo si presenta la struttura del VTO, seguito da una discussione sulla sua stabilità globali e locali; finalmente, è descritto una sintesi del VTO nel dominio discreto, dal momento che l'osservatore è tipicamente implementato in digitale sui microprocessori.

2.3.1. Struttura dell'osservatore

Nella figura 2.5 si mostra un diagramma a blocchi di un VTO in tempo continuo, l'osservatore insegue un generico vettore rotante $\vec{\Theta}$ e produce una stima della posizione angolare, o fase, e la velocità di ingresso. I quattro elementi costitutivi del VTO sono evidenziati:

- Vector cross-product (Prodotto vettoriale)
- Controllore
- Modello del sistema fisico
- Modello vettoriale

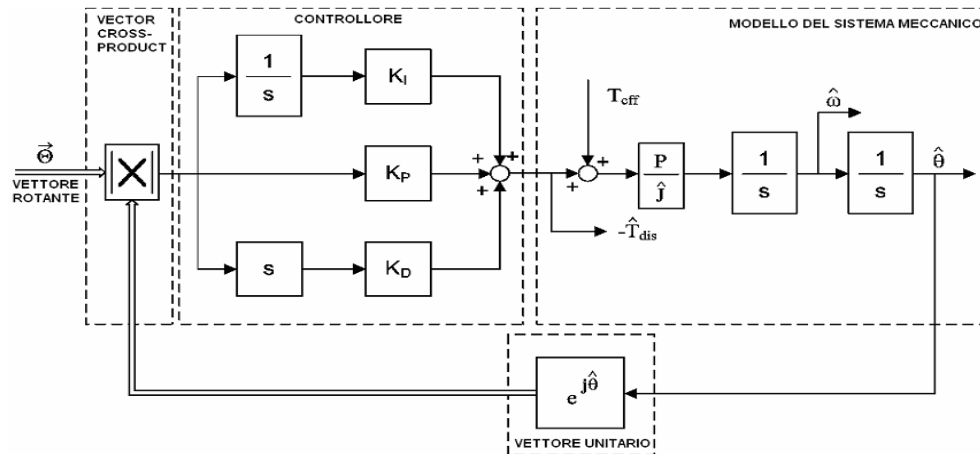


Fig. 2.5. Diagramma a blocchi del VTO

Il vector cross-product (Prodotto vettoriale) è il responsabile per la non-linearità dell'osservatore e per la sua natura di inseguimento vettoriale, il che lo rende molto simile ad un PLL. Il "controllore" è fondamentalmente un regolatore proporzionale, integrativo e derivativo (PID) ed è presente per forzare la convergenza del vettore stimato al vettore di ingresso; il vettore stimato è formato utilizzando la stima dello stato più esterno in combinazione con un adeguato modello vettoriale: nella figura 2.5 il modello vettoriale è stato costituito assumendo un ingresso ininterrottamente rotante. Finalmente, un modello del sistema fisico deve essere presente anche

perché un osservatore adeguatamente costituito dovrebbe produrre sempre una "coerente" stima dello stato fisico [Lor¹]: nella figura 2.5 si assume che il sistema fisico è un sistema meccanico rigidamente accoppiato, con smorzamento trascurabili, dove P è il numero di coppie polari della macchina e $\hat{J}$ è la stima dell'inerzia totale del sistema. Questo modello sarà utilizzato in tutto lo sviluppo della tesi. Un feedforward di ingresso può anche essere riconosciuto nel sistema fisico: un corretto feedforward aiuta a mantenere le proprietà di inseguimento sopra la larghezza di banda dell'osservatore. Inoltre, l'uscita del controllore può essere usata come una perturbazione che limita la larghezza di banda, come si vede nella fig. 2.5. La qualità di questa perturbazione dipende da molti fattori, tra cui la qualità del segnale feedforward, la larghezza di banda dell'osservatore ed anche il modello vettoriale di controreazione. Se il feedforward di ingresso non è presente, allora la struttura diventa quella di un filtro di stato e hanno ridotte proprietà di inseguimento sopra la larghezza di banda.

Una struttura alternativa che evita l'utilizzo del blocco derivativo è mostrata nella fig. 2.6. Le caratteristiche ingresso/uscita rimangono inalterate, ma ci sono due importanti modifiche interne. La prima è che, dal momento che l'azione derivativa è spostata allo stato di velocità, l'uscita del controllore PI non può essere considerata una perturbazione consistente. In secondo luogo, due diversi stati di velocità sono ora disponibili, uno che deriva direttamente dal controllore PI ed è indicato come "velocità non-migliorata", l'altra invece che contiene l'azione derivativa ed è quindi detta "velocità migliorata". Entrambe le stime di velocità possono essere utilizzate, sebbene la versione non-migliorata non è consistente con la stima della posizione ed i due segnali hanno diverse proprietà dinamiche. In particolare, il segnale migliorato è la stessa della stima della velocità nella fig. 2.5, il quale ha la stessa proprietà in termini di blocco dei disturbi; dall'altra parte, il segnale non-migliorato possiede ridotte proprietà di blocco delle perturbazioni, poiché è costituito senza il contributo del controllore derivativo.

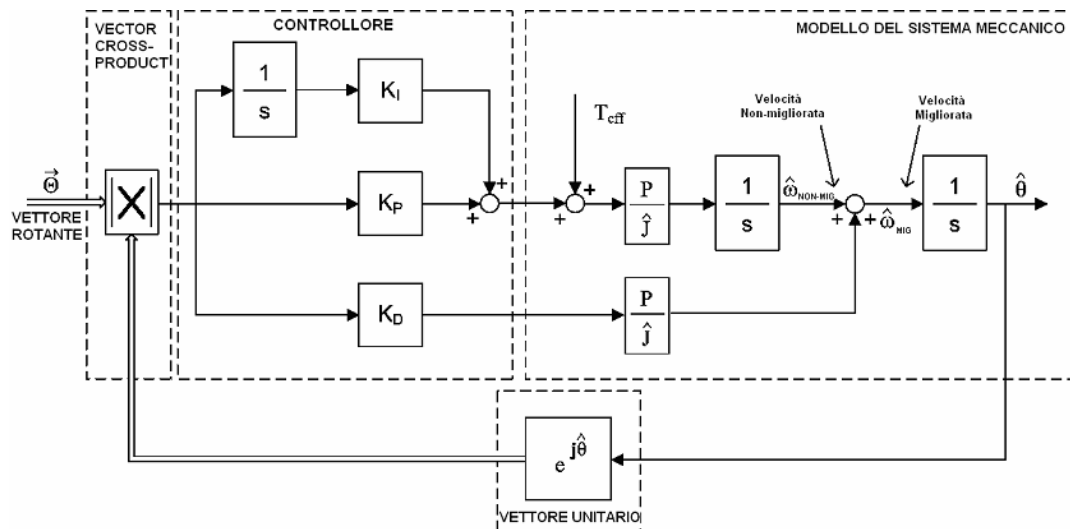


Fig. 2.6. Struttura alternativa del VTO

2.3.2. Analisi del punto di lavoro (linearizzazione dell'osservatore)

L'analisi del punto di lavoro del VTO è stato eseguita da [Deg¹]. Il modello del punto di lavoro è ottenuto utilizzando una approssimazione della Serie di Taylor di primo ordine intorno ad un specifico punto di lavoro ed i risultati ci portano ad un modello del punto di lavoro lineare descritto nella fig. 2.7.

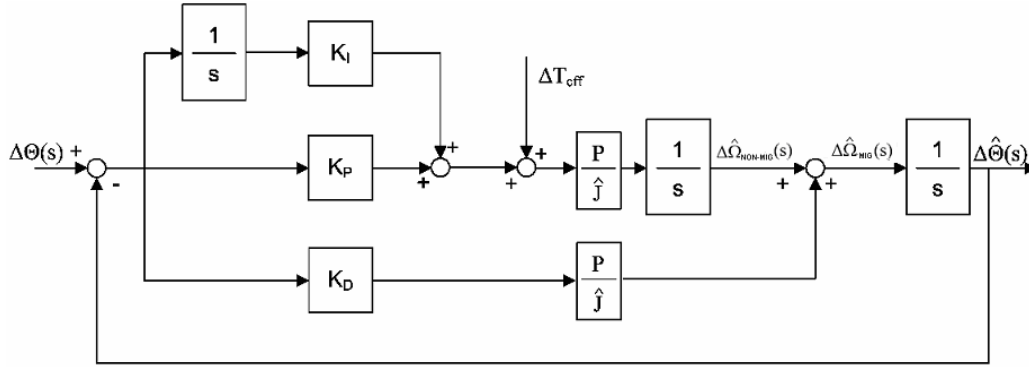


Fig. 2.7. Modello del punto di lavoro del VTO con la stima dell'errore molto piccolo

2.3.3. Sintesi dell'osservatore in tempo discreto

La sintesi del VTO in tempo discreto è stato eseguita da [DeD¹]. La fig. 2.8 mostra il modello del punto di lavoro del VTO in tempo discreto, topologicamente equivalente a quella della fig. 2.7; la sintesi dell'osservatore è realizzata per ottenere una desiderata risposta dinamica attraverso un appropriato valore delle tre guadagni dei controllori K_P , K_I e K_D . Ci sono altri parametri nel modello che devono essere scelti, cioè T_s e $\hat{J}$, questi devono essere selezionati a priori in quanto essi sono legati ai parametri fisici: la velocità con cui l'algoritmo del VTO è eseguito entro il controllore e l'inerzia totale del sistema meccanico. Per essere più precisi, è possibile variare il tempo di campionamento T_s , entro certi limiti, per vari motivi che vengono discussi, per esempio, in [Tes¹] e in [Giu¹]; comunque, c'è un limite minimo che è impostato per il sistema ed è pari al tempo di campionamento della PWM, oppure a metà del periodo di modulazione, a seconda della strategia della PWM adottata.

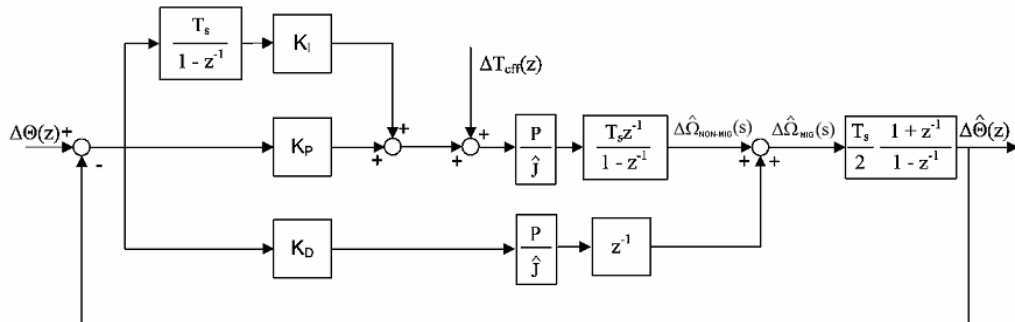


Fig. 2.8. Modello del punto di lavoro del VTO in tempo discreto

Al fine di semplificare i calcoli, la fig. 2.9 mostra il modello del punto di lavoro del VTO quando tutti i parametri e guadagni sono stati concentrati entro i quattro nuovi guadagni K_1 , K_2 , K_3 e K_{CFF} , le cui espressioni sono derivate nella eq. 2.10:

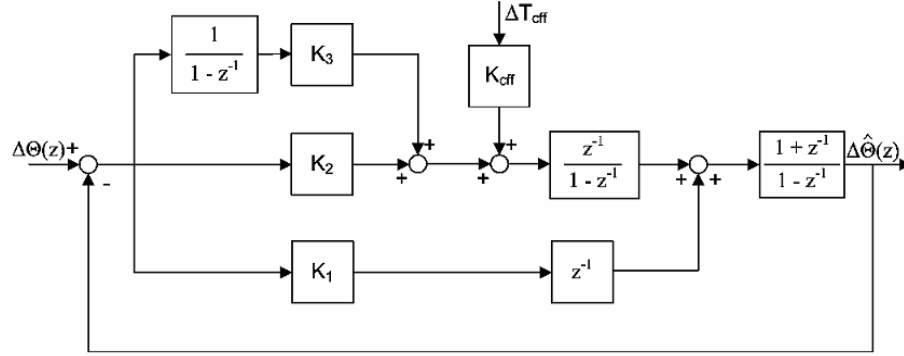


Fig. 2.9. Modello del punto di lavoro del VTO con i nuovi guadagni

$$(2.10) \quad \begin{cases} K_1 = \frac{K_D P T_S}{2\hat{J}} \\ K_2 = \frac{K_P P T_S^2}{2\hat{J}} \\ K_3 = \frac{K_I P T_S^3}{2\hat{J}} \\ K_{CFF} = \frac{P T_S^2}{2\hat{J}} \end{cases}$$

Inoltre, dal momento che il segnale di feedforward non ha alcuna influenza sul posizionamento dei poli dell'anello chiuso dell'osservatore, questo può essere escluso del modello del punto di lavoro del VTO visto nella fig. 2.9. La figura 2.10 mostra il modello risultante:

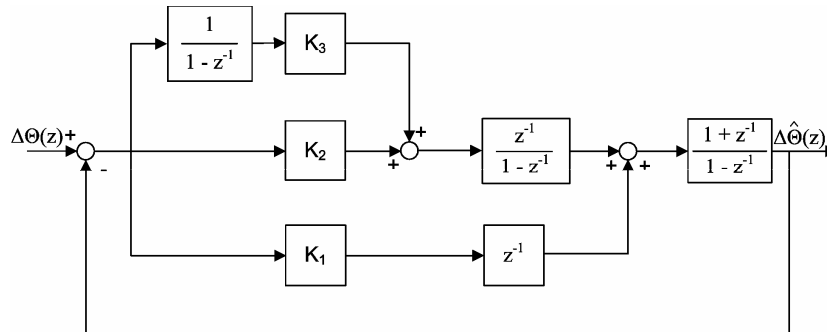


Fig. 2.10. Modello del punto di lavoro del VTO senza feedforward

Si può notare che i tre guadagni sono legati a tre distinti percorsi paralleli nel VTO; questa tipologia aiuta a semplificare l'analisi, se gli autovalori sono scelti per essere sufficientemente distanziati. La figura 2.11 aiuta a spiegare perché: essa mostra un

diagramma a blocchi equivalente in cui il percorso parallelo della precedente figura è stato manipolato algebricamente in modo di evidenziare i tre anelli in cascata del VTO.

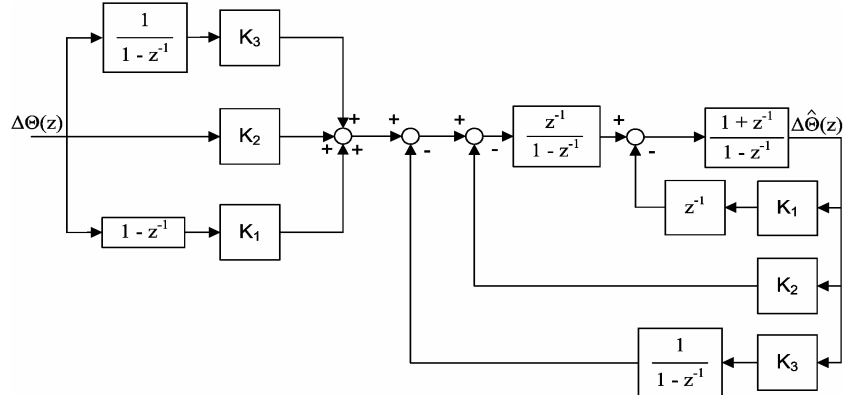


Fig. 2.11. Modello equivalente del punto di lavoro del VTO evidenziando l'anello derivativo, proporzionale e integrativo

Una semplice procedura, sarà ora presentata per tarare l'osservatore. Se l'anello derivativo è tarato per essere il più veloce dei tre anelli, può essere analizzato separatamente degli altri due anelli esterni.

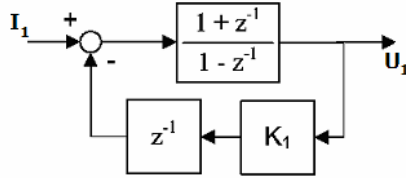


Fig. 2.12. Anello derivativo del modello del punto di lavoro del VTO

La funzione di trasferimento dell'anello derivativo mostrato in fig. 2.12 si presenta come:

$$(2.11) \quad F_1(s) = \frac{U_1}{I_1} = \frac{z^2 + z}{z^2 + (K_1 - 1)z + K_1}$$

dove l'equazione caratteristica è:

$$(2.12) \quad z^2 + (K_1 - 1)z + K_1 = 0$$

Al fine di impostare K_1 , il più piccolo tra gli zeri dell'equazione caratteristica (cioè il più veloce dei due poli ad anello chiuso) è libero di variare, mentre l'altro è impostato su z_1 , definito come:

$$(2.13) \quad z_1 = e^{-\frac{T_s}{\tau_1}}$$

dove τ_1 è la costante di tempo desiderata per l'anello derivativo. Invece K_1 è ottenuto dalle equazioni 2.12 e 2.13 come:

$$(2.14) \quad K_1 = z_1 \frac{1 - z_1}{1 + z_1}$$

Possiamo trovare l'intervallo di K_1 per cui il sistema è stabile, mediante la prova di stabilità di Jury [Oga¹] ed utilizzando l'eq. caratteristica (2.12) :

$$P(z) = z^2 + (K_1 - 1)z + K_1 \Rightarrow \begin{cases} a_0 = 1 \\ a_1 = (K_1 - 1) \\ a_2 = K_1 \end{cases}$$

Dato che si tratta di un sistema di secondo ordine, le condizioni di Jury possono scriversi come da seguito:

- 1-. $|a_2| < a_0$;
- 2-. $P(1) > 0$;
- 3-. $P(-1) > 0$.

Applicando ora la prima condizione di stabilità vediamo che:

$$|K_1| < 1 \Rightarrow -1 < K_1 < 1$$

La seconda condizione si converte in:

$$P(1) = (1)^2 + (K_1 - 1)(1) + K_1 > 0 \Rightarrow 1 + K_1 - 1 + K_1 > 0 \Rightarrow 2 \cdot K_1 > 0 \Rightarrow K_1 > 0$$

Per la terza condizione di stabilità abbiamo che:

$$P(-1) = (-1)^2 + (K_1 - 1)(-1) + K_1 > 0 \Rightarrow 1 - K_1 + 1 + K_1 > 0 \Rightarrow 2 > 0$$

Per stabilità, la costante di guadagno K_1 deve soddisfare le disuguaglianze 1, 2 e 3. Perciò,

$$0 < K_1 < 1$$

L'intervallo de la costante di guadagno K_1 per cui il sistema è stabile sarà tra 0 ed 1. Dunque la selezione di K_1 è limitata dal fatto che si devono evitare i poli complessi, allora il discriminante dell'eq. 2.12 non può essere negativo. Il limite massimo per il valore di K_1 per cui il sistema non abbia oscillazioni si trova uguagliando il discriminante dell'equazione caratteristica dell'anello derivativo a zero perché così siamo sicuri che i poli rimangono sull'asse reale.

$$P(z) = z^2 + (K_1 - 1)z + K_1 \Rightarrow \begin{cases} a = 1 \\ b = (K_1 - 1) \\ c = K_1 \end{cases} \Rightarrow \begin{aligned} &Discrim(P(z)) = (K_1 - 1)^2 - 4 \cdot 1 \cdot K_1 = 0 \\ &K_1^2 - 6 \cdot K_1 + 1 = 0 \Rightarrow \begin{cases} K_1 = 0.1716 \\ K_1 = 5.8284 \end{cases} \end{aligned}$$

Il limite massimo per il valore di K_1 è 0.1716 poiché, come si è detto sopra, il sistema è stabile per un K_1 tra 0 e 1. Questo valore di $K_1 = 0.1716$ corrisponde ad avere due poli coincidenti con $z = 0.4142$. La figura 2.13 mostra il luogo delle radici dell'anello derivativo del punto di lavoro del VTO e conferma l'esistenza del limite di K_1 oltre il quale il sistema diventa oscillatorio.

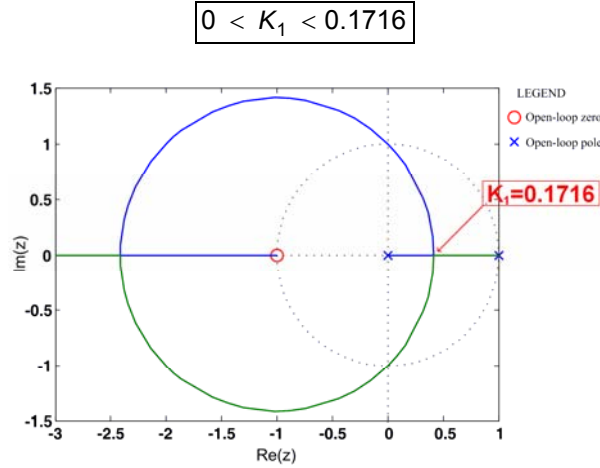


Fig. 2.13. Luogo delle radici dell'anello derivativo del punto di lavoro del VTO

Dopo aver tarato l'anello derivativo, è ora possibile procedere ad impostare l'anello proporzionale.

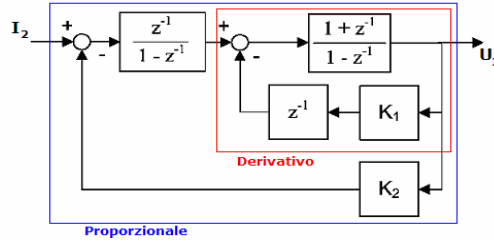


Fig. 2.14. Anello derivativo – proporzionale del modello del punto di lavoro del VTO

La funzione di trasferimento dell'anello proporzionale (incluendo l'anello derivativo) è:

$$(2.15) \quad F_2(s) = \frac{U_2}{I_2} = \frac{z^2 + z}{z^3 + (K_2 + K_1 - 2)z^2 + (K_2 + 1)z - K_1}$$

dove l'equazione caratteristica ha la seguente espressione:

$$(2.16) \quad z^3 + (K_2 + K_1 - 2)z^2 + (K_2 + 1)z - K_1 = 0$$

Il polo dell'anello proporzionale è impostato per essere z_2 pari a:

$$(2.17) \quad z_2 = e^{-\frac{T_s}{\tau_2}}$$

dove τ_2 è la costante di tempo desiderata per l'anello proporzionale. K_2 può essere ottenuta riordinando l'equazione 2.16:

$$(2.18) \quad K_2 = \frac{-z_2^3 + (2 - K_1)z_2^2 - z_2 + K_1}{z_2(z_2 + 1)}$$

Anche K_2 è limitata per evitare i poli complessi; in questo caso il limite dipende dalla scelta di K_1 . Ovviamente, se il guadagno scelto per l'anello derivativo è minore, anche il guadagno massimo per l'anello proporzionale sarà minore. Un esempio del luogo delle radici per l'anello derivativo – proporzionale è mostrato nella figura 2.15. Il punto di rottura che è visibile all'interno del cerchio unitario conferma l'esistenza del limite superiore di K_2 .

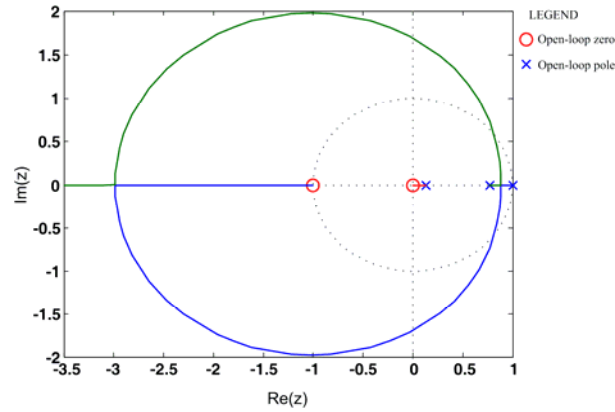


Fig. 2.15. Luogo delle radici dell'anello derivativo - proporzionale del punto di lavoro del VTO

Finalmente, si realizza la taratura dell'anello integrale. A tal fine, è ora necessario considerare l'intero sistema per ottenere la funzione di trasferimento:

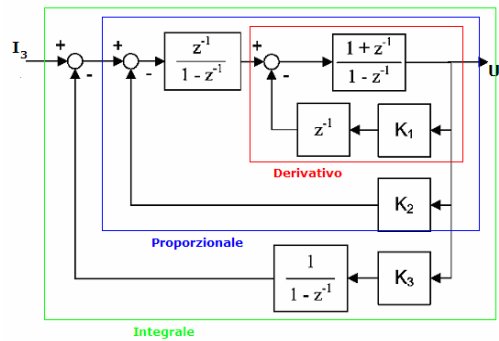


Fig. 2.16. Anello derivativo – proporzionale – integrale del modello del punto di lavoro del VTO

$$(2.19) \quad F_3(s) = \frac{U_3}{I_3} = \frac{z^3 + z^2}{z^4 + (K_3 + K_2 + K_1 - 3)z^3 + (K_3 - K_1 + 3)z^2 + (K_2 + K_1 + 1)z + K_1}$$

l'equazione caratteristica di quest'anello è:

$$(2.20) \quad z^4 + (K_3 + K_2 + K_1 - 3)z^3 + (K_3 - K_1 + 3)z^2 + (K_2 + K_1 + 1)z + K_1 = 0$$

Il polo dell'anello integrativo è impostato per essere z_3 uguale a:

$$(2.21) \quad z_3 = e^{-\frac{T_s}{\tau_3}}$$

dove τ_3 è la costante di tempo desiderata per l'anello integrale. K_3 può quindi essere ottenuta in base a:

$$(2.22) \quad K_3 = \frac{-z_3^4 - (K_2 + K_1 - 3)z_3^3 + (K_1 - 3)z_3^2 + (K_2 + K_1 + 1)z_3 - K_1}{z_3^2(z_3 + 1)}$$

K_3 pure è limitato, ed il limite dipende della scelta di K_1 e K_2 . Un esempio del luogo delle radici per l'anello derivativo – proporzionale – integrale è mostrato nella figura 2.17. E' necessaria una vista ingrandita per distinguere le dinamiche aggiunte dall'anello integrale. Il punto di rottura prossimo a $z = 1$ coincide con la scelta del valore limite di K_3 .

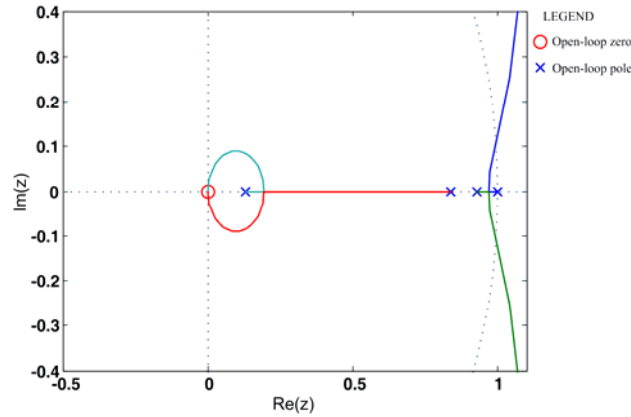


Fig. 2.17. Luogo delle radici per l'anello derivativo – proporzionale – integrale del punto di lavoro del VTO

Le equazioni 2.14, 2.18 e 2.22 possono essere utilizzate per la taratura del VTO se e solo se è verificata la seguente condizione:

$$(2.23) \quad \tau_3 \ll \tau_2 \ll \tau_1$$

Un fattore di almeno 10 tra le costanti di tempo è raccomandato. In ogni caso, siccome questo metodo di taratura è approssimativo, è consigliabile verificare il corretto posizionamento dei poli per il calcolo degli zeri dell'equazione caratteristica di tutto il sistema nell'eq. 2.20, e verificare se sono equivalenti agli zeri desiderati

dalle equazioni 2.13, 2.17 e 2.21. Di solito uno o più iterazioni di questa procedura sono necessarie per ottenere la larghezza di banda per tutti e tre gli anelli.

2.4. Applicazione dell'osservatore per un sistema di misura di posizione e velocità attraverso sonde ad effetto Hall

Il vettore Hall $\vec{H}_{\alpha\beta}$ è un vettore rotante quantizzato, quindi il VTO si comporterà in maniera diversa quando fa l'inseguimento, rispetto a quando il vettore di ingresso è in continua rotazione. Ciò è dovuto all'introduzione di una certa quantità di rumore di quantizzazione, che è formato da tutte le armoniche di ordine superiore evidenziate nell'equazione 2.9. L'effetto è una distorsione nella traiettoria del vettore stimato, che può causare un aumento significativo dell'errore di stima. Questo paragrafo descrive i problemi che sorgono nell'inseguimento del vettore Hall e propone alcune soluzioni che cercano di migliorare l'inseguimento delle prestazioni attraverso una riduzione delle armoniche all'ingresso che producono rumore.

2.4.1. Problemi di quantizzazione

Il VTO è progettato per inseguire la posizione angolare di ingresso, per cui se l'ingresso è quantizzato, il VTO tenderà ad inseguire il suo carattere quantizzato. La figura 2.18 chiarisce questo concetto: essa mostra una simulazione in regime stazionario della posizione reale, quantizzata e stimata quando la velocità angolare è di 10 rad/s elettrici e l'osservatore è impostato per avere come larghezze di banda 40, 4 e 0,4 Hz.

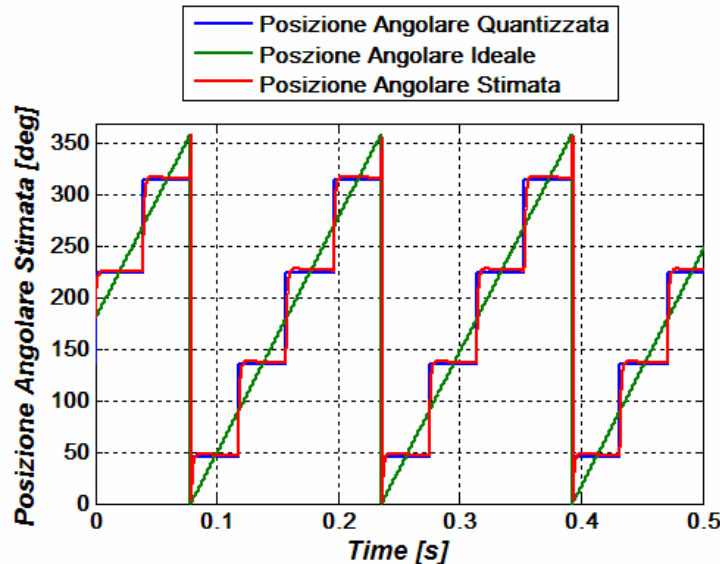


Fig. 2.18. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

Questo comportamento d'inseguimento erroneo è ulteriormente evidenziato nella figura 2.19, dove l'errore della posizione angolare stimata è comparata con un vettore di ingresso in continua rotazione. L'errore della posizione angolare stimata oscilla tra un ritardo di $\approx 37^\circ$ elettrici ed un anticipo di $\approx 43^\circ$ elettrici, mentre l'errore nel vettore di ingresso a continua rotazione è pari a zero. Ovviamente, tale stima di posizione angolare è inutile per essere utilizzata come riferimento per trasformazioni di coordinate.

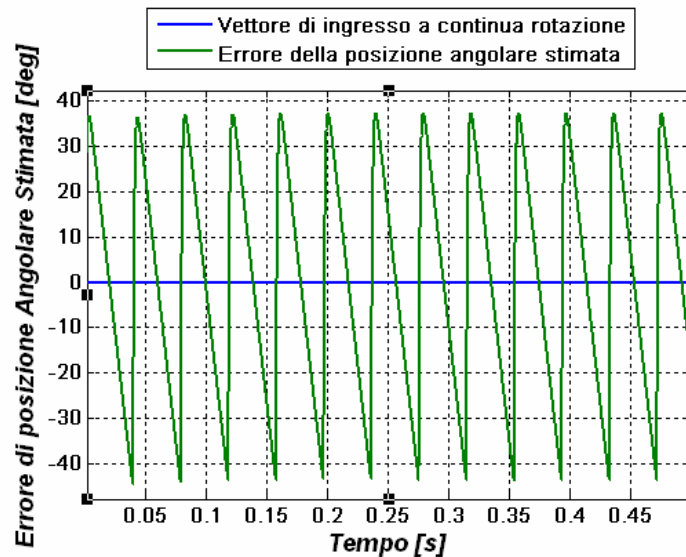


Fig. 2.19. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=10\text{rad/s}$; $BW = 40,4,0.4\text{Hz}$)

Lo stesso comportamento errato nell'inseguimento può essere visto nella velocità angolare stimata. La figura 2.20 mostra una simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario per le stesse condizioni delle precedenti figure. Una forte oscillazione può essere visualizzata nella velocità stimata dovuta al contributo dell'anello derivativo, permettendo che il rumore di quantizzazione presente all'ingresso sia portato quasi direttamente alla stima della velocità angolare, con un solo campione di tempo di ritardo. Viceversa, la velocità angolare stimata non-migliorata avrà una ridotta oscillazione poiché non contiene il termine derivativo ed il rumore di quantizzazione proveniente dall'ingresso è filtrato da una integrazione. La figura 2.21 mostra la simulazione della velocità angolare stimata non-migliorata con la stessa scala della figura 2.20, con l'intenzione di dimostrare la riduzione del rumore contenuto.

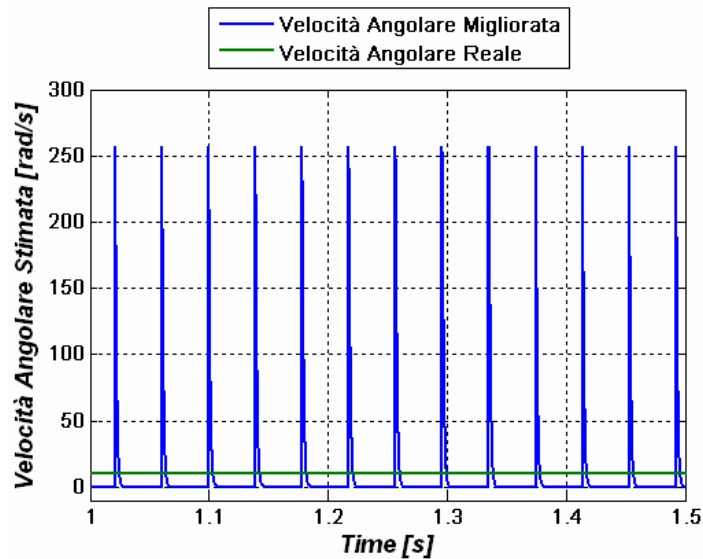


Fig. 2.20. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; $BW = 40,4,0.4$ Hz)

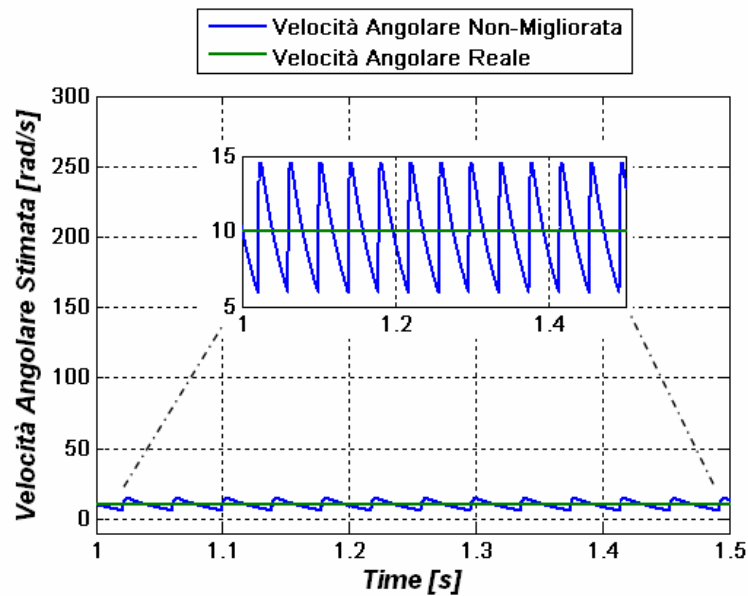


Fig. 2.21. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; $BW = 40,4,0.4$ Hz)

E' molto interessante analizzare gli stessi casi di prima utilizzando un incremento della velocità. La figura 2.22 mostra una simulazione in regime stazionario della posizione reale, quantizzata e stimata quando la velocità angolare è di 400 rad/s elettrici; l'osservatore è impostato per avere le stesse larghezze di banda. La figura 2.23 mostra l'errore della velocità angolare stimata. Si può vedere un migliore inseguimento per parte del VTO della posizione reale e l'oscillazione dell'errore della velocità angolare stimata è ridotta a circa un ottavo rispetto alla simulazione a 10 rad/s elettrici.

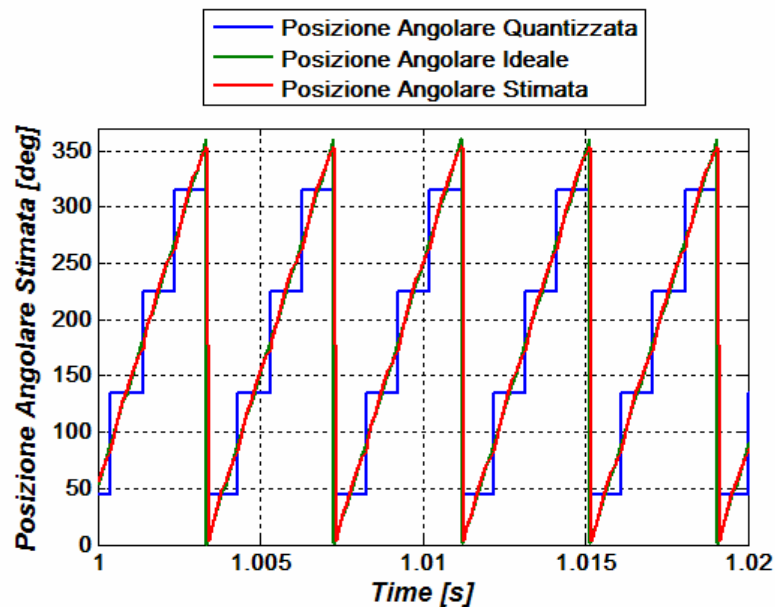


Fig. 2.22. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

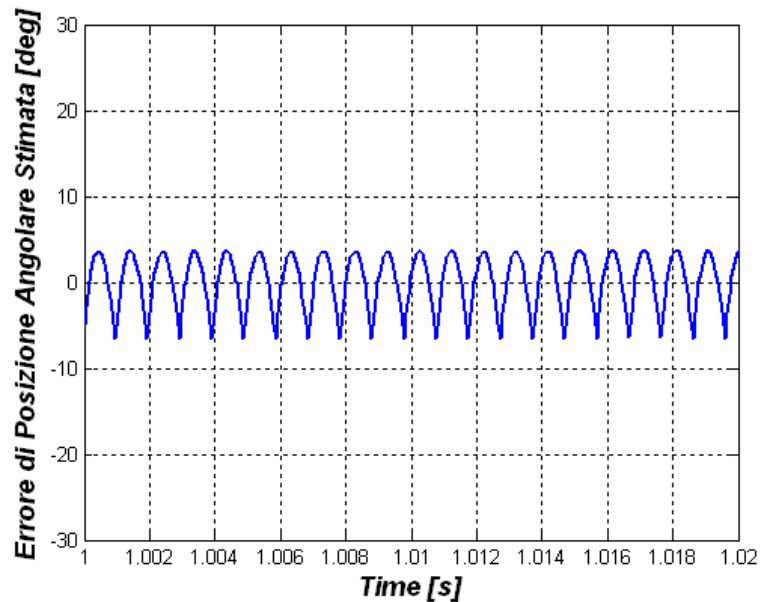


Fig. 2.23. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

Il motivo di questo comportamento è facilmente comprensibile confrontando le figure 2.18 e 2.22. Nella figura 2.20, il tasso con cui il vettore Hall cambia di stato, cioè la velocità media, è molto lento rispetto alla larghezza di banda dell'osservatore, quindi l'osservatore è in grado di inseguire la quantizzazione naturale dell'ingresso in maniera molto precisa; d'altra parte, nella figura 2.24 è evidente che la velocità angolare media del vettore Hall è incrementata in modo tale che l'osservatore non

risponde più velocemente come prima a cambiamenti improvvisi dell'ingresso quantizzato.

L'effetto vantaggioso delle velocità alte può essere visto nella stima della velocità angolare. La figura 2.24 mostra una simulazione della stima della velocità angolare migliorata a regime stazionario; questa fig. 2.24 rispetto alla figura 2.20 ha un rapporto tra l'oscillazione e la velocità reale ridotto in un fattore di quasi 30.

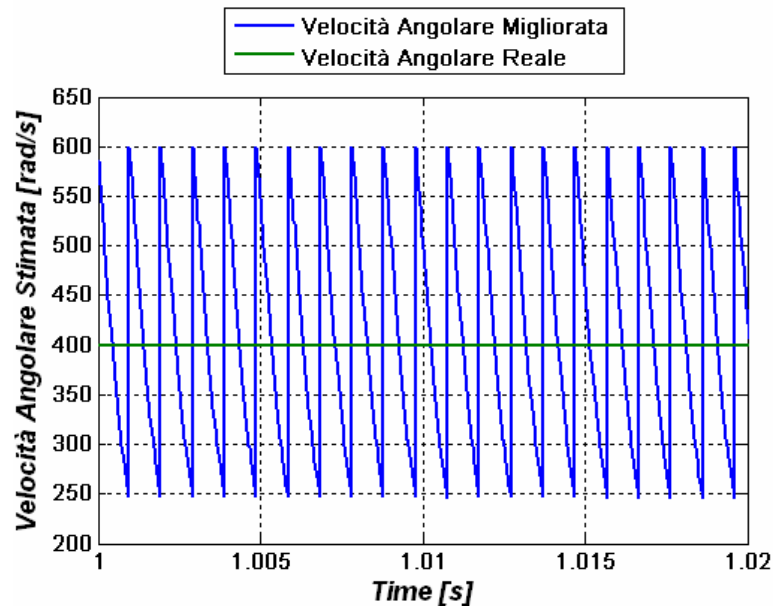


Fig. 2.24. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; $BW = 40,4,0.4$ Hz)

La figura 2.25 mostra la simulazione della velocità angolare stimata non-migliorata. Anche in questo caso, l'oscillazione è ridotta dimostrando tutti i vantaggi delle alte velocità. Si può concludere che il VTO migliora la sua proprietà di filtrare il rumore di quantizzazione all'aumentare della velocità.

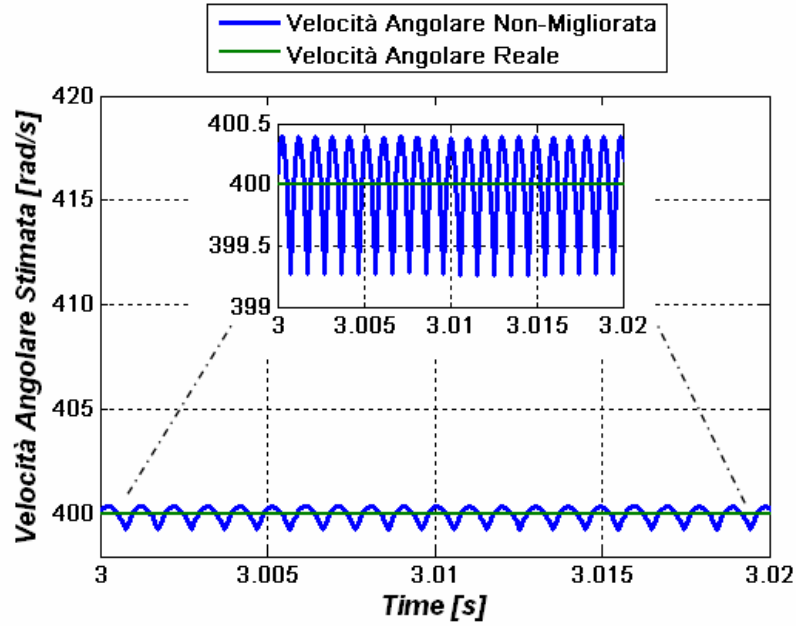


Fig. 2.25. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; $BW = 40,4,0.4$ Hz)

Più informazioni su questa importante proprietà dipendente dalla velocità si hanno attraverso l'analisi del periodo di campionamento della posizione del rotore. Il periodo di campionamento della posizione del rotore, T_{PS} , può essere definito come:

$$(2.24) \quad T_{PS} = \frac{\Delta\theta}{\omega_{el}} = \frac{2\pi}{N_{DS} \cdot \omega_{el}}$$

dove N_{DS} è il numero di stati discreti, pari a quattro in questo caso, e ω_{el} è la velocità angolare elettrica in rad/s. L'inverso di T_{PS} è uguale alla frequenza di campionamento della posizione del rotore f_{PS} :

$$(2.25) \quad f_{PS} = \frac{1}{T_{PS}} = \frac{N_{DS} \cdot \omega_{el}}{2\pi}$$

Se la velocità aumenta, f_{PS} aumenta proporzionalmente; la larghezza di banda dell'osservatore però rimane con le stesse caratteristiche. Maggiore è la frequenza di campionamento della posizione del rotore rispetto alla larghezza di banda del VTO, migliore sarà la proprietà di filtraggio di rumore armonico del VTO.

Al fine di formalizzare analiticamente questa ultima affermazione, un rapporto di campionamento SR, $[Har^1]$ e $[Har^2]$, può essere definita come:

$$(2.26) \quad SR = \frac{f_{PS}}{f_{BW}} = \frac{N_{DS} \cdot \omega_{el}}{2\pi \cdot f_{BW}}$$

dove f_{BW} è la larghezza di banda dell'osservatore in Hz. Il rapporto di campionamento indica quanto veloce è la frequenza di campionamento della posizione del rotore rispetto alla larghezza di banda del VTO. Per il caso della figura 2.20 il $SR = 0.1592$, mentre per il caso della figura 2.24 il $SR = 6,3662$. Ciò significa che nel primo caso la frequenza di campionamento del rotore è circa un quindicesimo della larghezza di banda, mentre nel secondo caso è circa sei volte la larghezza di banda. Affinché l'osservatore sia in grado di filtrare il rumore di quantizzazione efficacemente, la frequenza di campionamento del rotore deve essere almeno il doppio della larghezza di banda del VTO, cioè almeno pari alla frequenza di Nyquist.

L'importanza della soppressione del rumore armonico di quantizzazione è ulteriormente evidenziata, se le prestazioni di un sistema di controllo di movimento vengono analizzate quando l'osservatore è messo in controreazione.

2.4.2. Gain Scheduling

E' stato dimostrato nel paragrafo precedente che l'osservatore ha una naturale capacità di filtrare il rumore di quantizzazione all'aumentare della frequenza di campionamento SR della posizione d'ingresso. Una strategia di Gain Scheduling può migliorare questa proprietà di filtraggio regolando la larghezza di banda del VTO linearmente tra velocità zero e velocità massima, oltre la quale i guadagni sono tenuti pari al valore nominale; in questo modo la frequenza di campionamento è sempre mantenuta sufficientemente elevata con l'intenzione che l'osservatore possa filtrare effettivamente il rumore di quantizzazione. La figura 2.26 mostra il rapporto tra il generico guadagno dell'osservatore e la velocità. Dal momento che la velocità è calcolata dall'osservatore stesso, è possibile utilizzare per la Gain Scheduling una qualsiasi delle due velocità stimate. A causa della sua proprietà di filtraggio, la velocità angolare non-migliorata sarà la migliore scelta e sarà utilizzata in questa tesi.

Un avvertimento nella implementazione di questo Gain Scheduling è che il valore iniziale del guadagno non può essere impostato a zero a velocità zero: se così fosse, l'osservatore diventerebbe un osservatore ad anello aperto, solo con il termine di feedforward che fornisce l'informazione di velocità e posizione angolare stimata; teoricamente questo funziona, con la condizione che il modello del sistema fisico utilizzato per l'osservatore sia corretto. In pratica, siccome il modello del sistema fisico non è noto con precisione, si sa che occorre almeno un piccolo guadagno a bassa velocità. Come regola generale, un valore compreso tra 5% e 10% dei valori nominali può essere utilizzato a velocità zero.

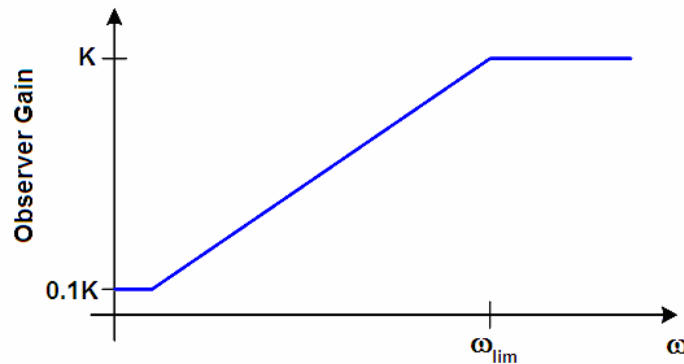


Fig. 2.26. Gain Scheduling del VTO in funzione della velocità angolare per limitare il rumore di quantizzazione

La velocità massima alla quale la larghezza di banda dell'osservatore è pari al valore nominale può essere ottenuta dalla frequenza di campionamento, definita nell'eq. 2.26. La tipica selezione del SR è di circa 8-10, ma può anche essere scelta per essere più grande se si desidera un extra-filtraggio del rumore di quantizzazione o inferiore se si desidera una regolazione più aggressiva, con un valore minimo di 2 che corrisponde al limite teorico imposto dalla frequenza di Nyquist. Riordinando l'eq. 2.26 si ottiene:

$$(2.27) \quad \omega_{lim} = \frac{2\pi \cdot f_{BW} \cdot SR}{N_{DS}}$$

Si sono realizzate simulazioni per verificare il miglioramento delle prestazioni quando la proposta della strategia di Gain Scheduling è implementata, con le stesse condizioni, cioè velocità angolare iniziale e larghezza di banda dell'osservatore nominale, che si sono realizzate nel paragrafo precedente. Il SR è stato fissato in 8, ottenendosi un valore di $\omega_{lim} = 503 \text{ rad/s}$.

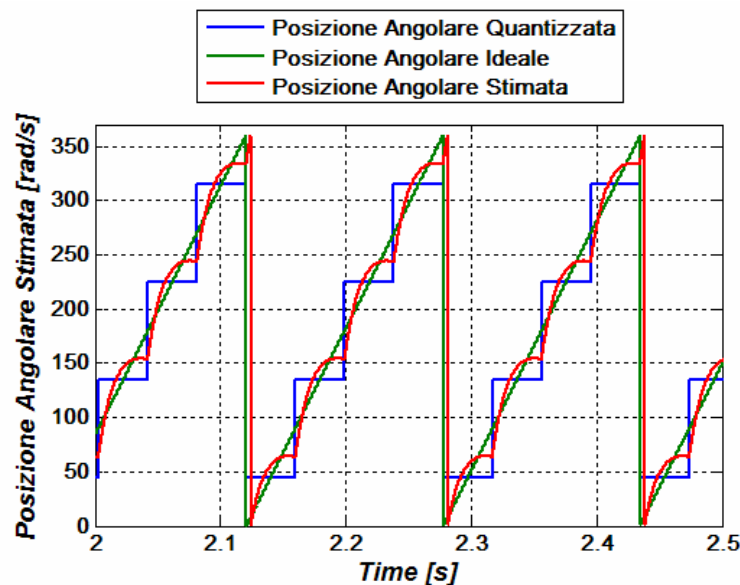


Fig. 2.27. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=10$ rad/s ; SR = 8 ; BW = 40,4,0.4 Hz)

La figura 2.27 mostra la simulazione in regime stazionario della posizione angolare reale, quantizzata e stimata, a 10 rad/s elettrici; rispetto alla figura 2.18 si può osservare che la larghezza di banda è stata notevolmente abbassata. L'errore di posizione angolare stimata della figura 2.28 mostra un'oscillazione di circa 40° elettrici, che rispetto all'oscillazione di 80° elettrici visti nella fig. 2.19, dimostra che l'ampiezza dell'oscillazione nell'errore di posizione angolare stimata è stata ridotta al 50%.

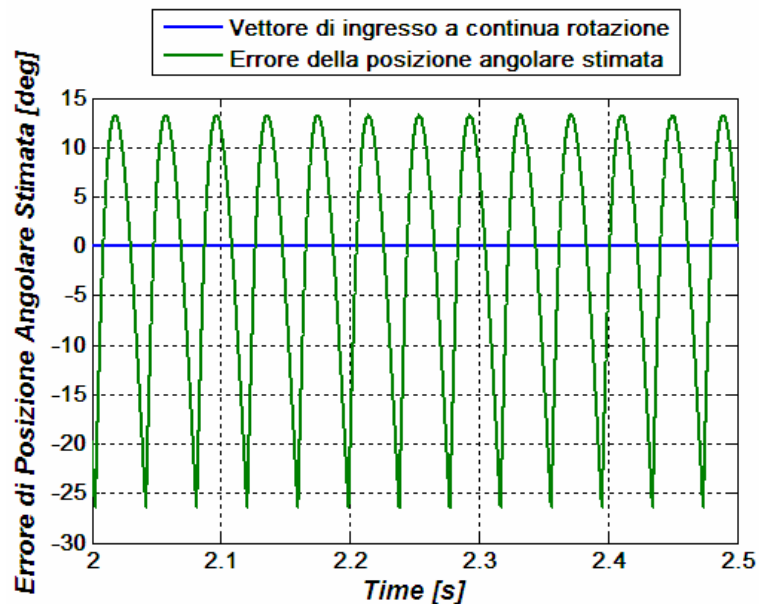


Fig. 2.28. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=10$ rad/s ; SR = 8 ; BW = 40,4,0.4 Hz)

Un miglioramento può essere osservato anche nella velocità angolare stimata. La fig. 2.29 mostra la simulazione della velocità angolare migliorata; si può notare che l'ampiezza dell'oscillazione è stata ridotta di un fattore di circa 10, rispetto a quella osservata nella figura 2.20. La velocità angolare non-migliorata è mostrata nella fig. 2.30 e può essere paragonata con quella osservata nella fig. 2.21: il segnale oscilla tra 7 e 12 rad/s elettrici, anziché tra 6 e 15 rad/s. Il miglioramento non è così significativo per questa stima rispetto alla stima nel caso migliorato. Ciò è dovuto al fatto che il segnale non-migliorato è meno influenzato dal rumore di quantizzazione.

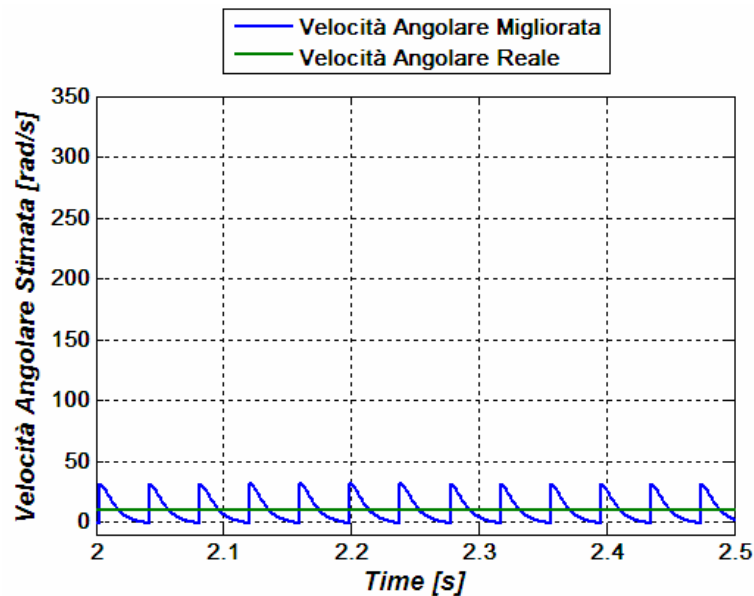


Fig. 2.29. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; $SR = 8$; $BW = 40,4,0.4$ Hz)

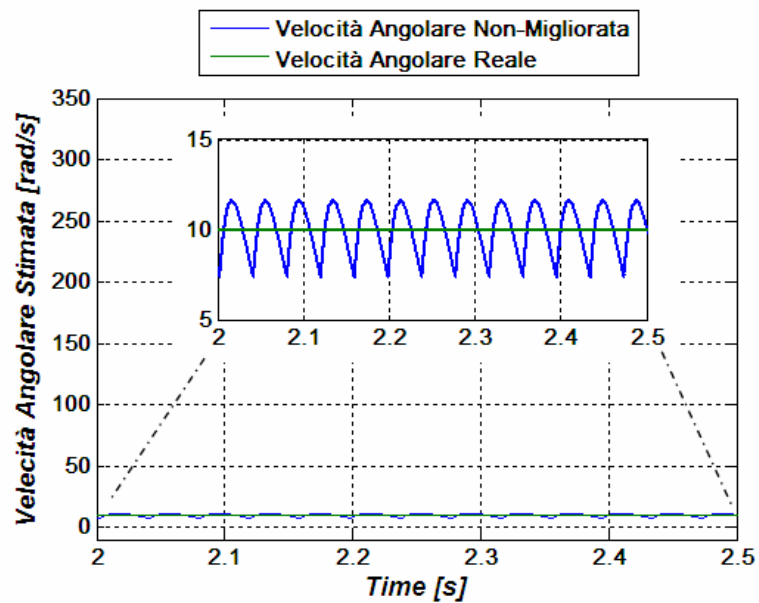


Fig. 2.30. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; $SR = 8$; $BW = 40,4,0.4$ Hz)

Le prestazioni sono state esaminate anche a 400 rad/s elettrici; le figure 2.31 e 2.32 mostrano un miglioramento rispetto alle figure 2.22 e 2.23 in cui non è usato la Gain Scheduling. L'errore di posizione angolare stimata è notevolmente inferiore a 5° elettrici, sufficientemente basso per essere utilizzato come riferimento per trasformazione di coordinate.

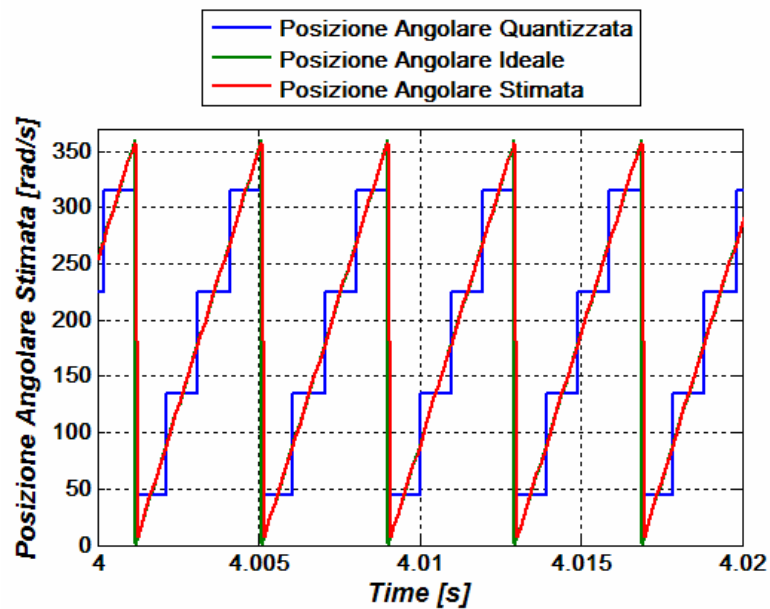


Fig. 2.31. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=400\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$)

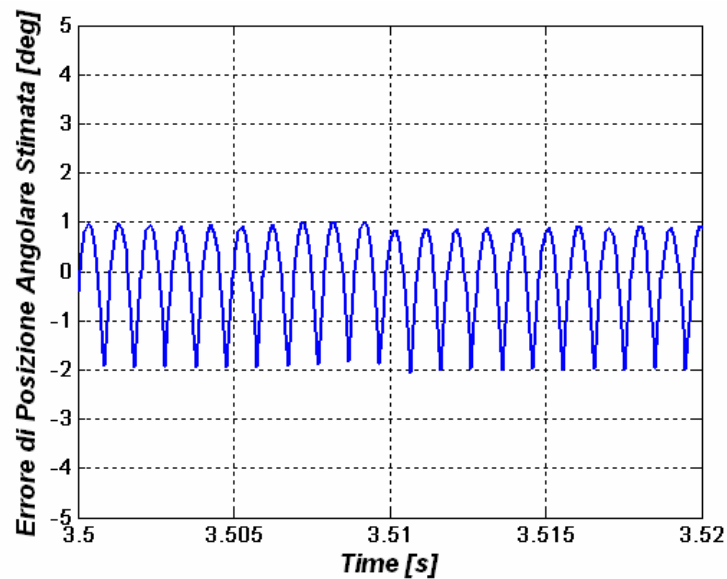


Fig. 2.32. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=400\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$)

Le figure 2.33 e 2.34 mostrano la velocità angolare migliorata e non-migliorata; rispetto alle loro controparti delle figure 2.24 e 2.25 un miglioramento nell'oscillazione è ottenuto soprattutto nel segnale migliorato, nel quale l'ampiezza dell'oscillazione è ridotta di circa il 35%.

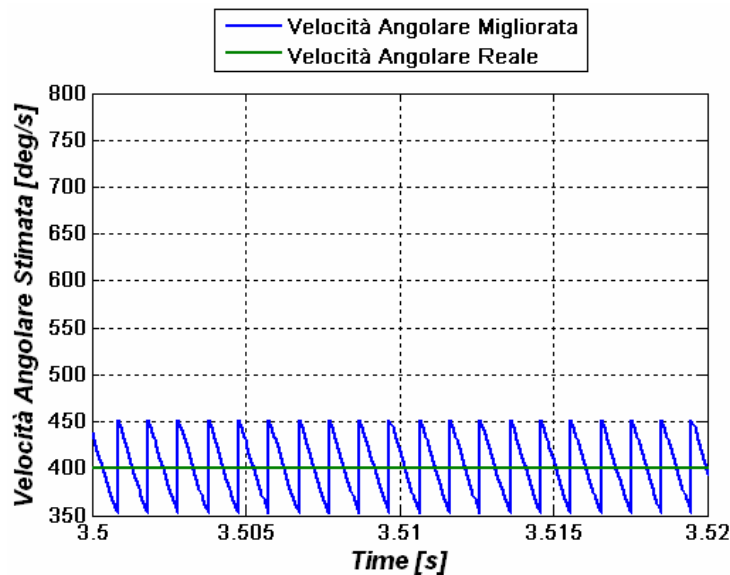


Fig. 2.33. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; $SR = 8$; $BW = 40,4,0.4$ Hz)

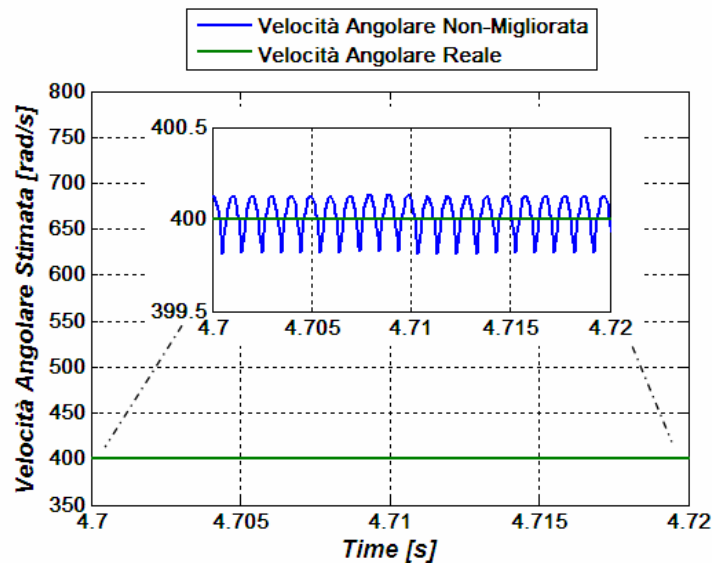


Fig. 2.34. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; $SR = 8$; $BW = 40,4,0.4$ Hz)

2.4.3. Disaccoppiamento delle armoniche di quantizzazione

Le tecniche di disaccoppiamento delle armoniche sono state analizzate in [Deg¹] e successivamente estese ai vettori quantizzati in [Tes¹]. L'idea di base è molto semplice: se il contenuto armonico di un vettore rotante è noto, è possibile disaccoppiarlo sottraendolo al vettore rotante stesso, al fine di ottenere una versione "pulita" costituita solo della componente fondamentale. Nel caso del Vettore Hall quantizzato $\vec{H}_{\alpha\beta}$, l'analisi armonico fatto nel paragrafo 2.2 mostra la Serie di Fourier

Spaziale di un vettore Hall rotante quantizzato nell'eq. 2.9. Il contenuto armonico che è la causa del rumore armonico di quantizzazione nel VTO e quindi deve essere disaccoppiato dal Vettore Hall, è:

$$(2.28) \quad \vec{H}_{\alpha\beta_armonico}(\theta_{el}) = \sum_{k=1}^{+\infty} \left[-\frac{1}{(4k-1)} e^{-j\left((4k-1)\theta_{el} + \frac{\pi}{4}\right)} + \frac{1}{(4k+1)} e^{j\left((4k+1)\theta_{el} - \frac{\pi}{4}\right)} \right]$$

La figura 2.35 mostra come tale disaccoppiamento è implementato nel VTO. Si forma un anello esterno che utilizza la posizione angolare stimata dall'osservatore per formare un Vettore Armonico Rotante $\vec{H}_{\alpha\beta_dec}$, che viene poi sottratto al Vettore Hall Rotante di ingresso. Il Vettore Risultante è il nuovo ingresso per il VTO.

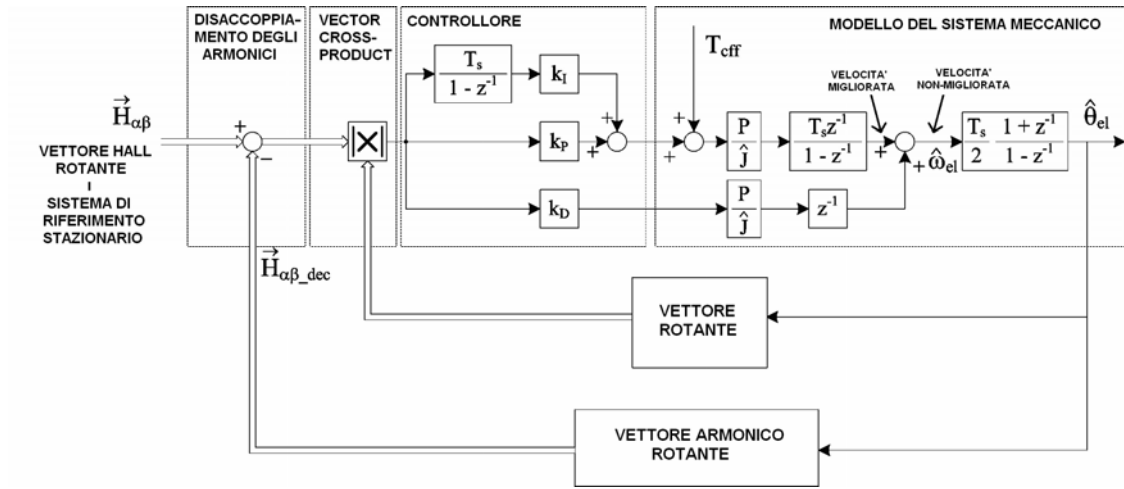


Fig. 2.35. Diagramma a blocchi del VTO utilizzando il Disaccoppiamento degli Armonici

Varie strategie di disaccoppiamento sono state studiate in [Tes¹], [Har¹] e [Har²]. Esse possono essere sostanzialmente raggruppate in 3 categorie:

1. Disaccoppiamento armonico troncato;
2. Disaccoppiamento armonico completo;
3. Versione senza ritardo del disaccoppiamento armonico completo filtrato.

Il disaccoppiamento armonico troncato consiste nel formare un modello armonico in cui solo un certo numero di armoniche sono considerate; ed esempio, se solo la 3^a e la 5^a armonica sono considerate, che corrisponde ai primi elementi della serie dell'eq. 2.28, allora il modello armonico è:

$$(2.29) \quad \vec{H}_{\alpha\beta}(\theta_{el}) = \frac{1}{3} e^{-j\left(3\theta_{el} + \frac{\pi}{4}\right)} - \frac{1}{5} e^{-j\left(5\theta_{el} - \frac{\pi}{4}\right)}$$

Se si considerano anche i due successivi termini della serie, cioè le 3^a, 5^a, 7^a e 9^a armoniche, allora il modello armonico diventa:

$$(2.20) \quad \vec{H}_{\alpha\beta}(\theta_{el}) = \frac{1}{3} e^{-j\left(3\theta_{el} + \frac{\pi}{4}\right)} - \frac{1}{5} e^{-j\left(5\theta_{el} - \frac{\pi}{4}\right)} + \frac{1}{11} e^{-j\left(11\theta_{el} + \frac{\pi}{4}\right)} - \frac{1}{13} e^{-j\left(13\theta_{el} - \frac{\pi}{4}\right)}$$

Il disaccoppiamento armonico completo richiede che l'intero contenuto armonico deve essere eliminato dal Vettore Hall; in questo caso il modello armonico può essere impostato come:

$$(2.21) \quad \vec{H}_{\alpha\beta_dec_full}(\theta_{el}) = \vec{H}_{\alpha\beta} - e^{j\left(\theta_{el} + \frac{\pi}{4}\right)}$$

Il modello del disaccoppiamento armonico completo è caratterizzato per avere una forte discontinuità ai transitori dei sensori; questo può essere visto nella figura 2.36 dove le forme d'onda di disaccoppiamento sono simulate rispetto al sistema di riferimento stazionario $\alpha\beta$. Queste discontinuità, causate dalla sommatoria di tutte le componenti armoniche più elevate, possono creare disaccoppiamenti sbagliati se sono presenti alcuni errori di stima iniziali, come ad esempio durante il transitorio dell'accelerazione. Il Dr. M.C. Harke propone in [Har¹] un modo per ridurre questo effetto con la creazione di una versione senza ritardo del disaccoppiamento armonico completo filtrato. Questo filtro senza ritardo può essere fatto in vari modi; un metodo efficiente è filtrare l'onda, dopodichè invertirla e filtrarla nuovamente. La forma d'onda risultante ha un comportamento più morbido all'intorno del transitorio, senza distorsioni significative rispetto al modello di disaccoppiamento completo intorno alla parte centrale del settori dei sensori; per questo motivo viene chiamati modello di forma d'onda di disaccoppiamento filtrato.

Il modo più semplice per implementare le strategie di disaccoppiamento è nel sistema di riferimento stazionario $\alpha\beta$; le forme d'onda di disaccoppiamento armonico sono calcolate e memorizzate in una look-up table; durante il funzionamento, questi valori vengono sottratti dai componenti H_α e H_β mostrati nella figura 2.2 per ottenere forme d'onda che hanno maggiore somiglianza con forme d'onde sinusoidali pure. La figura 2.36 mostra la forma d'onda di disaccoppiamento armonico nel caso di disaccoppiamento completo e disaccoppiamento armonico filtrato; intorno ad un forte transitorio, le due forme d'onda diventano praticamente identiche.

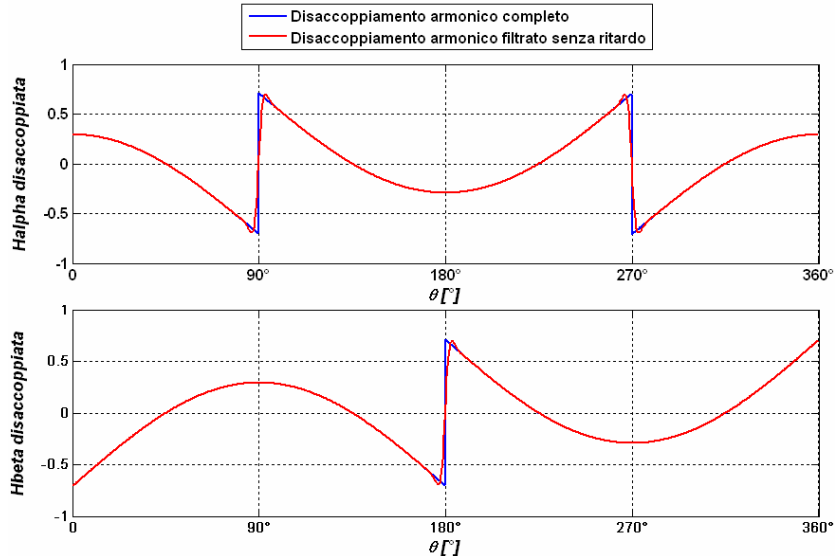


Fig. 2.36. Forma d'onda di disaccoppiamento armonico

La figura 2.37 mostra le componenti del sistema di riferimento stazionario come risultato della simulazione di $\vec{H}_{\alpha\beta} - \vec{H}_{\alpha\beta_dec}$, quando è utilizzata la forma d'onda disaccoppiata e senza disaccoppiamento. Alcuni errori residui possono essere osservati nella transizione degli stati dei sensori a causa dell'imperfetto disaccoppiamento.

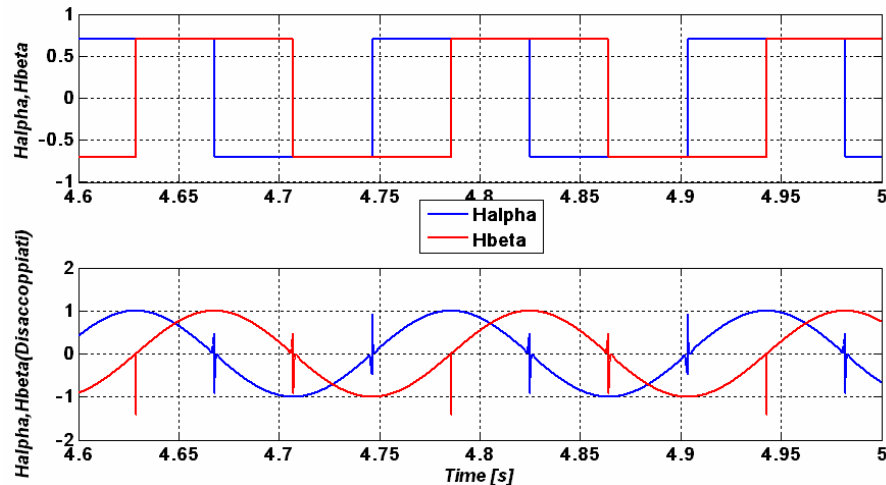


Fig. 2.37. Componenti del sistema di riferimento stazionario, utilizzando forma d'onda disaccoppiata

Si sono eseguite simulazioni in regime stazionario per 10 rad/s e 400 rad/s quando si utilizzano le tecniche di disaccoppiamento in combinazione con la Gain Scheduling descritta in precedenza, in modo da essere in grado di valutare il miglioramento nel VTO rispetto alle implementazioni descritte in precedenza.

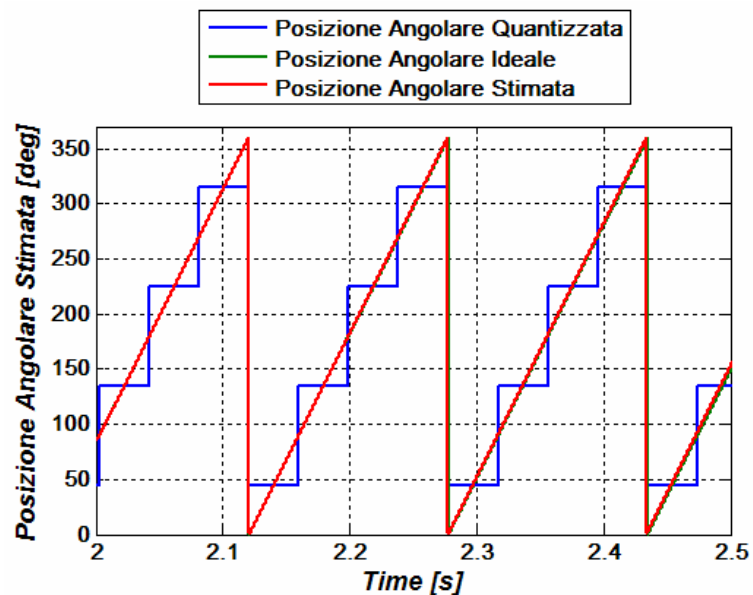


Fig. 2.38. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=10\text{rad/s}$; $SR=8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

La figura 2.38 mostra la simulazione in regime stazionario della posizione quantizzata, reale e stimata a 10 rad/s elettrici; rispetto alla figura 2.27, non ci sono oscillazioni visibili. L'errore della posizione angolare stimata può essere visto nella figura 2.39 essendo sempre minore ai 2° elettrici. Si tratta di un risultato significativo rispetto alle oscillazioni di quasi 80° elettrici che sono presenti quando non vengono utilizzati la Gain Scheduling ed il disaccoppiamento degli armonici.

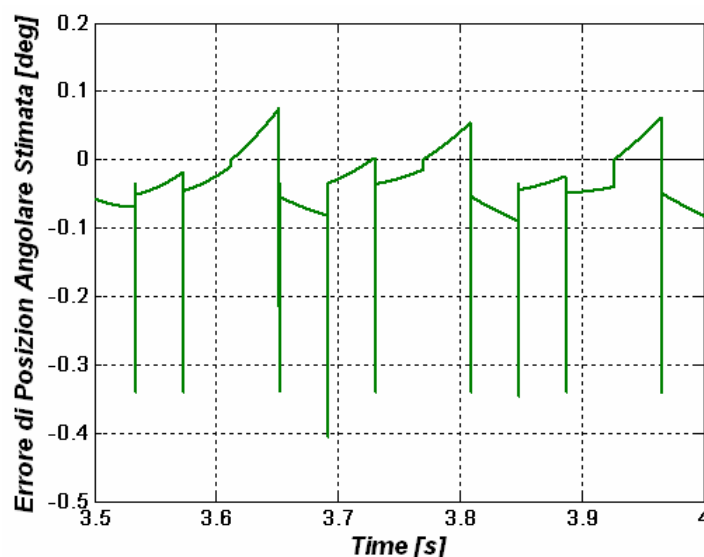


Fig. 2.39. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_e=10\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

Le figure 2.40 e 2.41 mostrano il comportamento in regime stazionario della velocità angolare stimata quando la velocità angolare reale è di 10 rad/s. La Velocità Angolare Migliorata possiede oscillazioni leggermente superiore rispetto alla figura 2.29, dove il disaccoppiamento degli armonici non è usato. Ciò è dovuto all'imperfetto disaccoppiamento che introduce alla transizione degli stati dei sensori. Tuttavia, questo effetto è molto meno evidente nella Velocità Angolare Non-Migliorata della figura 2.41, in cui l'oscillazione a causa del rumore armonico di quantizzazione è ridotto di un fattore circa 10 rispetto alla fig. 2.30.

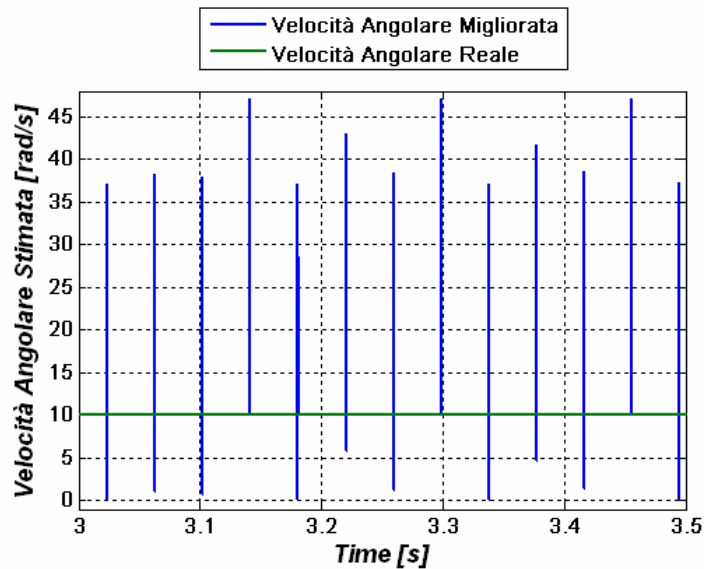


Fig. 2.40. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=10\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

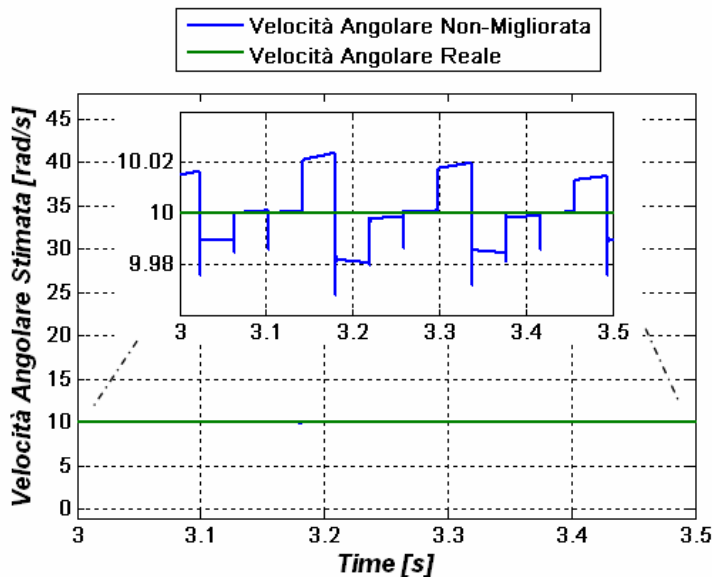


Fig. 2.41. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=10\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

Le figure 2.42, 2.43, 2.44 e 2.45 mostrano i risultati relativi al disaccoppiamento armonico quando la velocità reale è pari a 400 rad/s. Tutte le simulazioni in regime stazionario migliorano il suo comportamento rispetto alle simulazioni quando solo si utilizza la Gain Scheduling, eccetto nel caso della Velocità Angolare Migliorata mostrata nella figura 2.44 nella quale l'errore di disaccoppiamento non permette significativi miglioramenti.

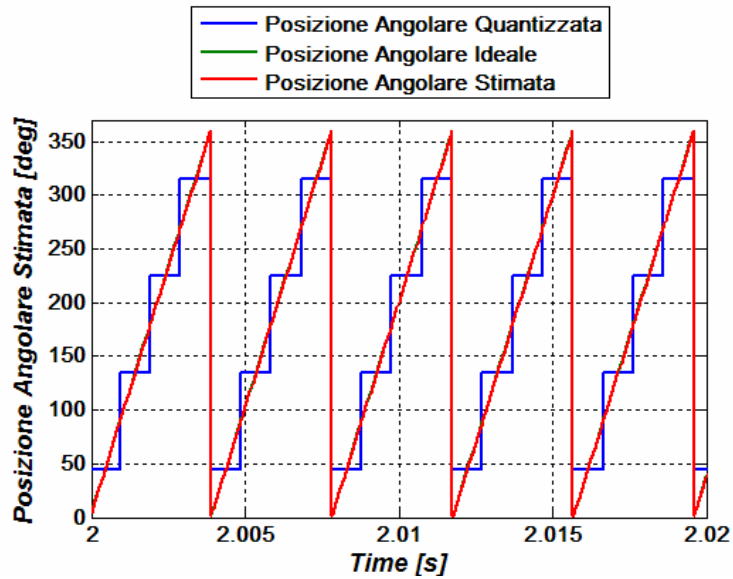


Fig. 2.42. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=400\text{rad/s}$; $SR=8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

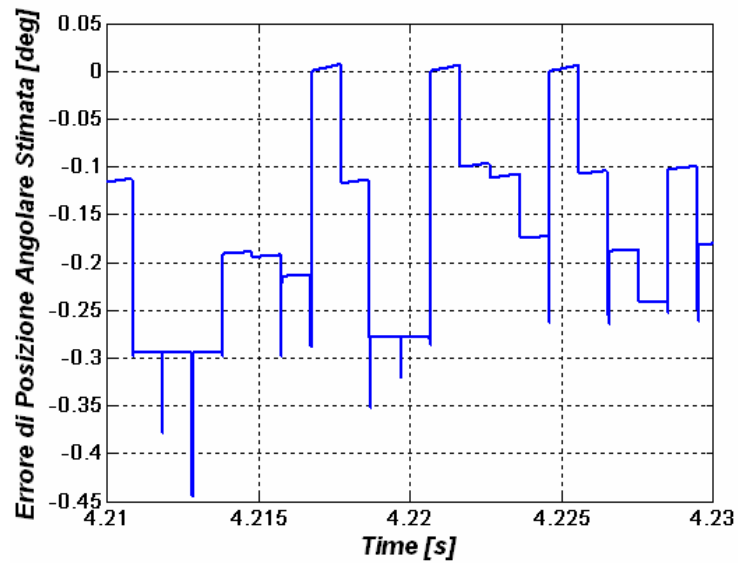


Fig. 2.43. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=400\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

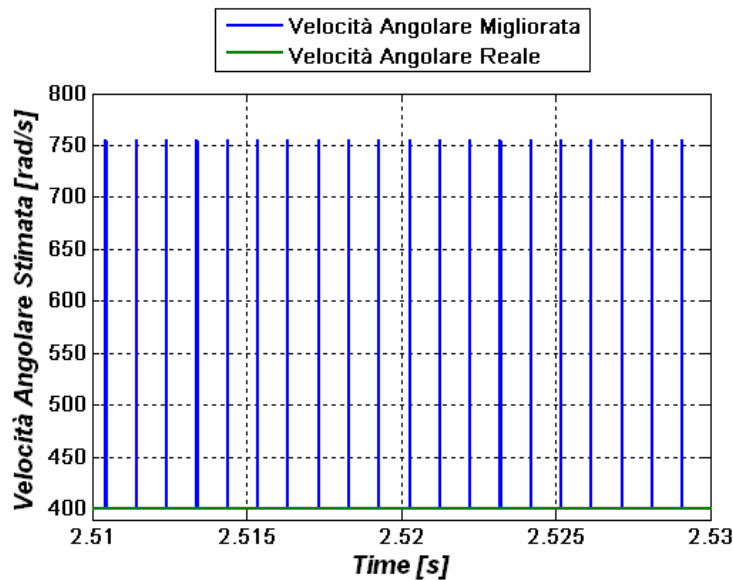


Fig. 2.44. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=400\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

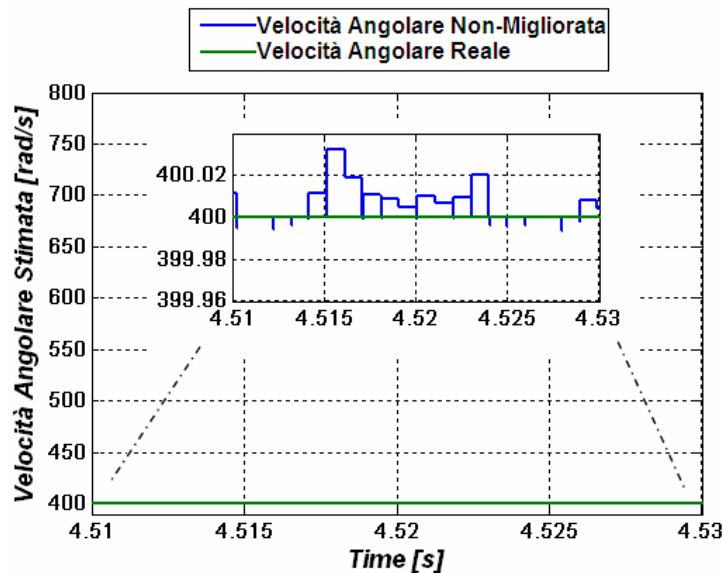


Fig. 2.45. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=400\text{rad/s}$; $SR = 8$; $BW = 40,4,0.4\text{Hz}$; Forma d'onda disaccoppiata)

La stabilità locale del disaccoppiamento armonico nel VTO può essere determinato allo stesso modo come si fa per il VTO senza utilizzare il disaccoppiamento, indagando sul prodotto vettoriale. Questo analisi è stata condotta dal Dr. T. R. Tesch e dall'Ing. Giulio De Donato.

Il prodotto vettoriale, ε , può essere ottenuto dai componenti del sistema di riferimento stazionario “ $\alpha\beta$ ” secondo la seguente equazione:

$$(2.21) \quad \varepsilon = \left| \vec{H}_{\alpha\beta} - \vec{H}_{\alpha\beta_dec} \right| \times e^{j\hat{\theta}_{el}} \\ \varepsilon = \left[H_{\beta}[\theta_{el}] - H_{\beta_dec}[\hat{\theta}_{el}] \right] \cdot \cos(\hat{\theta}_{el}) - \left[H_{\alpha}[\theta_{el}] - H_{\alpha_dec}[\hat{\theta}_{el}] \right] \cdot \sin(\hat{\theta}_{el})$$

$H_{\alpha_dec}[\hat{\theta}_{el}]$ e $H_{\beta_dec}[\hat{\theta}_{el}]$ sono selezionati secondo la strategia di disaccoppiamento scelta. L'equazione 2.21 può essere simulata in funzione dell'errore della posizione angolare stimata $\Delta\theta_{err}$. Se non sono presenti disaccoppiamenti, ε sarà simile ad una sinusoide, proprio come un semplice software di base Phase-Locked Loop (PLL). Il disaccoppiamento cambia la forma d'onda attesa, ma ε si comporta in modo simile. Inoltre, ε dipende anche della posizione angolare effettiva θ_{el} . Infatti, la seguente identità può scritta:

$$(2.22) \quad \Delta\theta_{err} = \theta_{el} - \hat{\theta}_{el} \Rightarrow \hat{\theta}_{el} = \theta_{el} - \Delta\theta_{err}$$

Sostituendo l'eq. 2.22 nella eq. 2.21 si arriva alla seguente espressione per ε , la quale evidenzia la sua dipendenza da entrambi i termini, l'errore di posizione angolare stimata e la posizione angolare reale:

$$(2.23) \quad \varepsilon = \left[H_{\beta}[\theta_{el}] - H_{\beta_dec}[\theta_{el} - \Delta\theta_{err}] \right] \cdot \cos(\theta_{el} - \Delta\theta_{err}) - \\ + \left[H_{\alpha}[\theta_{el}] - H_{\alpha_dec}[\theta_{el} - \Delta\theta_{err}] \right] \cdot \sin(\theta_{el} - \Delta\theta_{err})$$

La figura 2.46 è una simulazione della superficie che rappresenta l'eq. 2.23 nello spazio definito da $\Delta\theta_{err}$, θ_{el} e ε , quando si utilizza il disaccoppiamento completo. La curva tracciata nella figura 2.47 è la proiezione della figura 2.46 nel piano (ε , $\Delta\theta_{err}$) per $\theta_{el} = 45^\circ$, 135° , ecc., cioè nelle transizioni degli stati dei sensori. Quando l'errore di posizione angolare stimata $\Delta\theta_{err}$ è positivo, cioè la posizione angolare stimata è in ritardo rispetto alla posizione angolare reale, l'osservatore continuerà ad inseguire perché la polarità del prodotto vettoriale produce una controrazione negativa. Il punto di attraversamento limite per la stabilità per un errore di stima positivo si verifica nella figura 2.47 ai 135° : al di là di questo punto la ε si inverte di polarità, causando instabilità locale, cioè controeazione positiva.

Quando $\Delta\theta_{err}$ è negativo, cioè la posizione angolare stimata è in ritardo rispetto alla posizione angolare reale, ε rimane pari a zero fino ad avere un $\Delta\theta_{err}$ pari a -90° . Questo intervallo verrà chiamato “zona zero di ε ” nella seguente discussione. Esso è equivalente ad un errore inosservabile a causa della quantizzazione della controeazione, [Tes¹]. Per grandi errori di stima il prodotto vettoriale produce una controeazione negativa; in questa regione, il punto di attraversamento limite per la stabilità si verifica ai -225° .

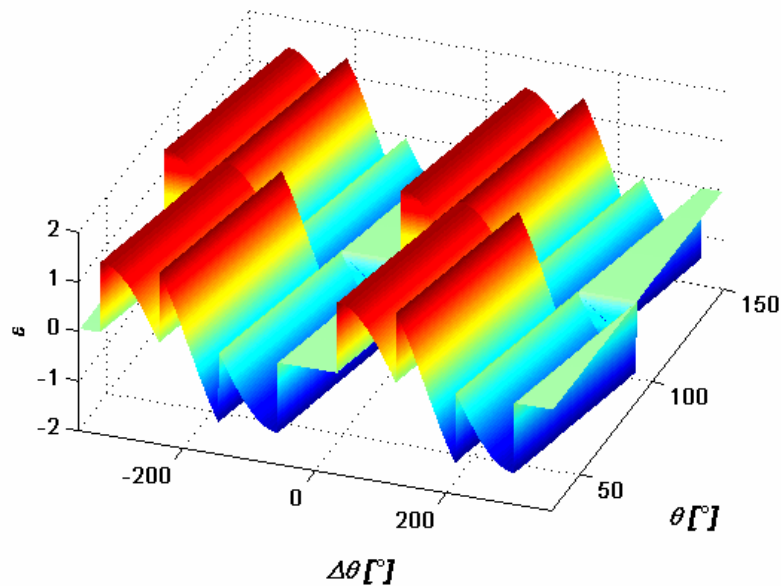


Fig. 2.46. Prodotto Vettoriale in funzione di $\Delta\theta_{err}$ e θ_{el}

La figura 2.46 mostra un'altra proprietà del prodotto vettoriale: per crescenti valori della posizione angolare reale θ_{el} , la proiezione sul piano $(\varepsilon, \Delta\theta_{err})$ ha la stessa forma d'onda, ma questa si sposta progressivamente nel verso positivo di $\Delta\theta_{err}$. Ad ogni transizione degli stati dei sensori, cioè $\theta_{el} = 90^\circ$ nella fig. 2.46, la forma d'onda riprende lo stesso aspetto della figura 2.47. In pratica, la “zona zero di ε ” si sposta linearmente per 45° in ogni settore in modo tale che l'osservatore possa diventare più sensibile ai piccoli errori di anticipo della stima e meno sensibile ai piccoli ritardi negli errori di stima. Poco prima della fine di ogni settore, la “zona zero di ε ” sarà contenuta nella parte positiva di $\Delta\theta_{err}$ del piano $(\varepsilon, \Delta\theta_{err})$. Inoltre, anche il punto di attraversamento limite per la stabilità si sposta linearmente di 45° in ogni settore: ad esempio, il punto di attraversamento limite per la stabilità per errori di stima positivi si sposta dai 135° all'inizio del settore ai 180° immediatamente prima della successiva transizione. Questa proprietà di spostamento è presente anche se non è usato il disaccoppiamento armonico, cioè quando ε è una funzione sinusoidale, perché è direttamente connessa col vettore Hall quantizzato $\vec{H}_{\alpha\beta}$.

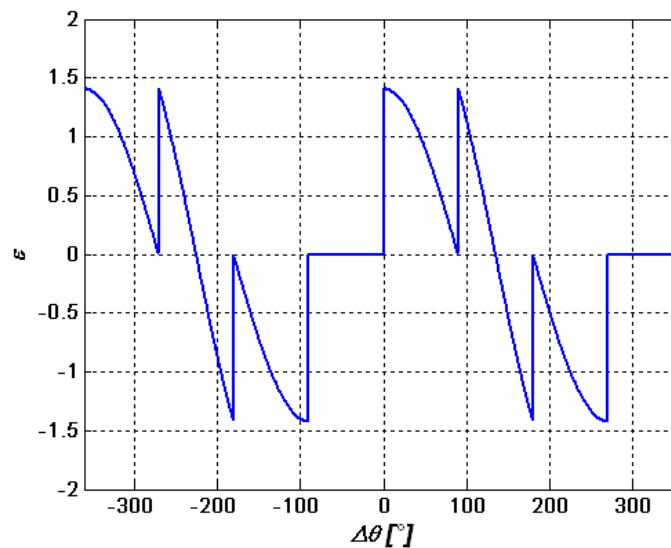


Fig. 2.47. Prodotto Vettoriale rispetto all'errore di posizione angolare stimata

2.5. Conclusioni

Questo capitolo analizza l'algoritmo per l'inseguimento della posizione angolare e velocità angolare stimata attraverso l'Osservatore ad inseguimento vettoriale (Vector Tracking Observer VTO). E' stata introdotta la linearizzazione del punto di lavoro, e poi l'analisi in tempo discreto del VTO per l'inseguimento del Vettore Hall con il quale si può ottenere la larghezza di banda dell'osservatore. I problemi di quantizzazione che si verificano quando è utilizzato il VTO per inseguire il vettore Hall sono descritti. E' stato anche dimostrato che la velocità con cui l'osservatore insegue il vettore Hall dipende dalle caratteristiche di filtraggio e questa proprietà è stata sfruttata per miglioramenti dell'inseguimento attraverso la Gain Scheduling (i guadagni dipendono dalla velocità). Inoltre, sono state descritte diverse strategie di disaccoppiamento armonico. In tal modo, con l'uso contemporaneamente della Gain Scheduling e con il disaccoppiamento delle armoniche, si insegue effettivamente la posizione e velocità angolare con basso errore di stima. Tutte queste analisi sono state fatte in Matlab; nel successivo capitolo si mettono alla prova queste analisi utilizzando il DSP eZdspTM (8) TMS320F2812.

⁸ eZdsp is a trademark of Spectrum Digital Inc (www.spectrumdigital.com)

Capitolo 3

Hardware e software del Digital Signal Processor (DSP) per l'implementazione dell'osservatore

3.1. Introduzione

In questo capitolo si descrive l'hardware ed il software del DSP (Code Composer Studio™⁽⁹⁾ – Platinum Edition) per l'implementazione dell'osservatore per misurare la posizione e velocità angolare. L'intero codice è in linguaggio C/C++ e assembler, il quale viene trattato per il software detto in precedenza nel DSP eZdsp™⁽¹⁰⁾ TMS320F2812. Il paragrafo 3.2 descrive l'hardware del DSP eZdsp™⁽¹⁰⁾ TMS320F2812, caratteristiche tecniche [TeR¹] e struttura [Dat¹]. Nel paragrafo 3.3 si presenta le periferiche utilizzate per l'implementazione dell'osservatore e una spiegazione dettagliata di tutte le principali routine utilizzate per il sistema di controllo, particolarmente quelle riguardanti il Vector Tracking Observer VTO. Per ultimo, il paragrafo 3.4 descrive le conclusioni intorno al software utilizzato per l'implementazione dell'osservatore.

3.2. Digital Signal Processors DSP (eZdsp™ TMS320F2812)

Il dispositivo eZdsp™ TMS320F2812 è un membro della generazione dei DSP TMS320C28x; sono processori dei segnali digitali altamente integrati e consentono soluzioni ad alte prestazioni per applicazioni in sistemi di controllo. Il dispositivo eZdsp™ TMS320F2812 sarà in alcuni casi abbreviato come F2812.

3.2.1. Caratteristiche tecniche del DSP

- | | |
|--|--|
| <ul style="list-style-type: none">• High-Performance Static CMOS Technology<ul style="list-style-type: none">- 150 MHz (6.67-ns Cycle Time)- Low-Power (1.8-V Core @135 MHz, 1.9-V Core @150 MHz, 3.3-V I/O) Design• JTAG Boundary Scan Support⁽¹¹⁾• High-Performance 32-Bit CPU (TMS320C28x)<ul style="list-style-type: none">- 16 x 16 and 32 x 32 MAC Operations | <ul style="list-style-type: none">- 16 x 16 Dual MAC- Harvard Bus Architecture- Atomic Operations- Fast Interrupt Response and Processing- Unified Memory Programming Model- 4M Linear Program/Data Address Reach- Code-Efficient (in C/C++ and Assembly)- TMS320F24x/LF240x Processor Source Code Compatible |
|--|--|

⁹ Code Composer Studio™ is a trademark of Texas Instruments (www.ti.com)

¹⁰ eZdsp is a trademark of Spectrum Digital Inc (www.spectrumdigital.com)

¹¹ IEEE Standard 1149.1-1990, IEEE Standard Test-Access Port

Trademark are the property of their respective owners.

- **On-Chip Memory**
 - Flash Devices: Up to 128K x 16 Flash (Four 8K x 16 and Six 16K x 16 Sectors)
 - ROM Devices: Up to 128K x 16 ROM
 - 1K x 16 OTP ROM
 - L0 and L1: 2 Blocks of 4K x 16 Each Single-Access RAM (SARAM)
 - H0: 1 Block of 8K x 16 SARAM
 - M0 and M1: 2 Blocks of 1K x 16 Each SARAM
- **Boot ROM (4K x 16)**
 - With Software Boot Modes
 - Standard Math Tables
- **External Interface (2812)**
 - Up to 1M Total Memory
 - Programmable Wait States
 - Programmable Read/Write Strobe Timing
 - Three Individual Chip Selects
- **Clock and System Control**
 - Dynamic PLL Ratio Changes Supported
 - On-Chip Oscillator
 - Watchdog Timer Module
- **Three External Interrupts**
- **Peripheral Interrupt Expansion (PIE) Block That Supports 45 Peripheral Interrupts**
- **Three 32-Bit CPU-Timers**
- **128-Bit Security Key/Lock**
 - Protects Flash/ROM/OTP and L0/L1 SARAM
 - Prevents Firmware Reverse Engineering
- **Motor Control Peripherals**
 - Two Event Managers (EVA, EVB)
 - Compatible to 240xA Devices
- **Serial Port Peripherals**
 - Serial Peripheral Interface (SPI)
 - Two Serial Communications Interfaces (SCIs), Standard UART
- Enhanced Controller Area Network (eCAN)
- Multichannel Buffered Serial Port (McBSP)
- **12-Bit ADC, 16 Channels**
 - 2 x 8 Channel Input Multiplexer
 - Two Sample-and-Hold
 - Single/Simultaneous Conversions
 - Fast Conversion Rate: 80 ns/12.5 MSPS
- **Up to 56 General Purpose I/O (GPIO) Pins**
- **Advanced Emulation Features**
 - Analysis and Breakpoint Functions
 - Real-Time Debug via Hardware
- **Development Tools Include**
 - ANSI C/C++ Compiler/Assembler/Linker
 - Code Composer Studio IDE
 - DSP/BIOS
 - JTAG Scan Controller[†]
- **Low-Power Modes and Power Savings**
 - IDLE, STANDBY, HALT Modes Supported
 - Disable Individual Peripheral Clocks
- **Package Options**
 - 179-Ball MicroStar BGA With External Memory Interface (GHH), (ZHH) (2812)
 - 176-Pin Low-Profile Quad Flatpack (LQFP) With External Memory Interface (PGF) (2812)
 - 128-Pin LQFP Without External Memory Interface (PBK) (2810, 2811)
- **Temperature Options:**
 - A: -40°C to 85°C (GHH, ZHH, PGF, PBK)
 - S: -40°C to 125°C (GHH, ZHH, PGF, PBK)
 - Q: -40°C to 125°C (PGF, PBK)

3.2.2. Struttura del DSP

Il DSP eZdsp™ F2812 è composto di quattro grandi blocchi interfaccia:

- Analog Interface Connector
- I/O Interface Connector
- JTAG Interface
- Parallel Port JTAG Controller Interface

La figura 3.1 mostra la configurazione di base per il DSP F2812 e la figura 3.2 mostra il suo diagramma a blocchi funzionale. Le principali interfacce del DSP sono: “JTAG interface” ed “Expansion interface”.

TMS320C24x, Code Composer Studio, DSP/BIOS, and MicroStar BGA are trademarks of Texas Instruments.

[†] IEEE Standard 1149.1–1990, IEEE Standard Test-Access Port

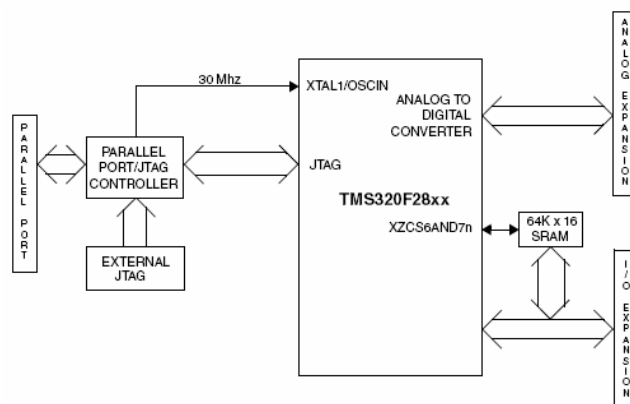


Fig. 3.1. Configurazione di base del DSP eZdsp™ F2812 [Dat¹]

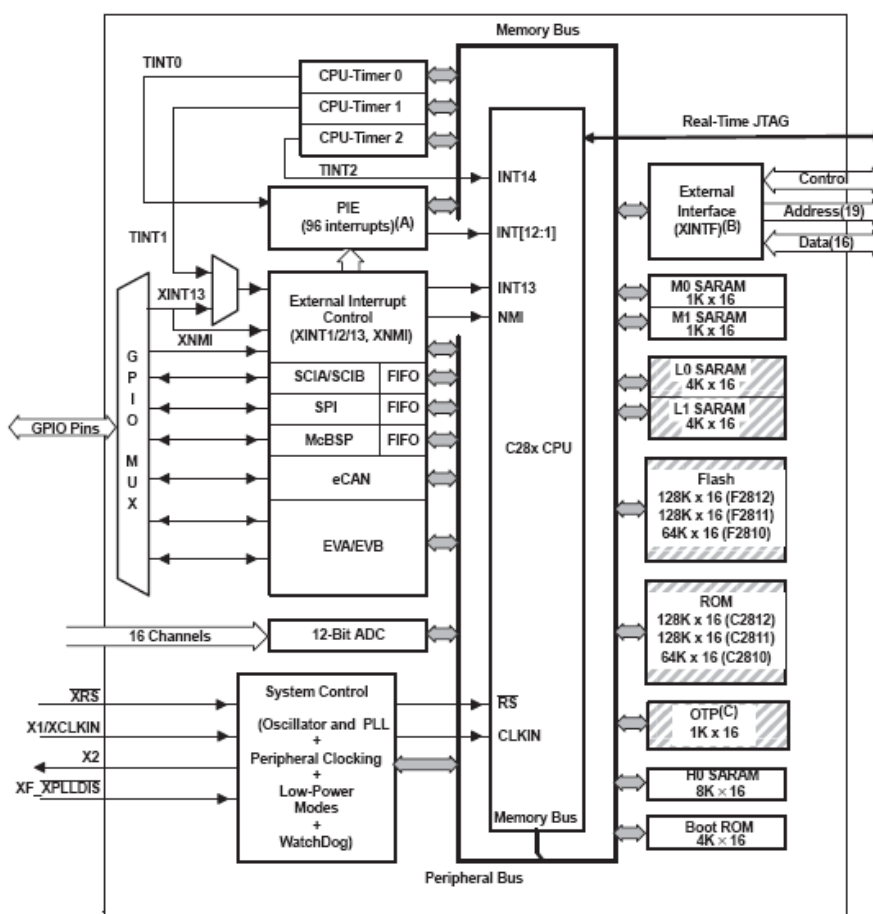


Fig. 3.2. Diagramma a blocchi funzionale del eZdspTM F2812 [Dat¹] ⁽¹³⁾

13 Protected by the code-security module.

NOTES:

- A. 45 of the possible 96 interrupts are used on the devices.
B. XINTF is available on the F2812 and C2812 devices only.
C. On C281x devices, the OTP is replaced with a 1K X 16 block of ROM

3.2.2.1. eZdsp™ F2812 Scheda

La scheda del eZdsp™ F2812 è un circuito multistrato stampato con dimensioni pari a 5.25 x 3.0 pollici, alimentato a +5V da un alimentatore esterno. La figura 3.3 mostra la distribuzione dei componenti di entrambe le versioni (socketed e unsocketed) della eZdsp™ F2812.

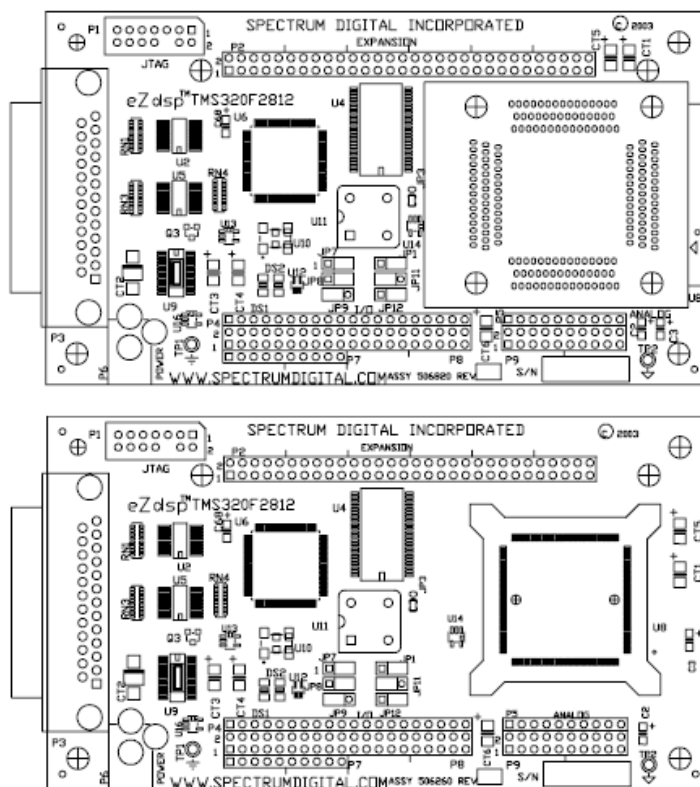


Fig. 3.3. Versione (socketed e unsocketed) della eZdsp™ F2812 [Dat¹]

3.2.2.1.1. Alimentazione

La scheda del eZdsp™ F2812 è alimentata da un alimentatore a +5V, inclusa nella unità. La unità richiede una corrente da 500mA, la alimentazione è fornita tramite il connettore P6. Se sono collegati al eZdsp™ F2812 delle schede di espansione, sarà necessario alimentare con una maggiore corrente.

3.2.2.2. eZdsp™ F2812 Memoria

Il eZdsp™ F2812 include all'interno le seguenti tipi di memoria:

- 128K x 16 Flash
- 2 blocchi di 4K x 16 Single Access RAM (SARAM)
- 1 blocco di 8K x 16 SARAM
- 2 blocchi di 1K x 16 SARAM

In aggiunta è fornita un'altra memoria esterna di 64K x 16 SRAM . Il processore sulla eZdsp™ F2812 può essere configurato nella modalità boot loader o non-boot loader. Il eZdsp™ F2812 può caricare la RAM per il debug o può essere caricata la FLASH ROM ed eseguito. Per grandi progetti di software si propone di fare un iniziale debug con il modulo eZdsp™ F2812, che supporta un totale ambiente di RAM.

3.2.2.2.1. Mappa della memoria

La figura 3.4 mostra la mappa della memoria che si trova sul eZdsp™ F2812:

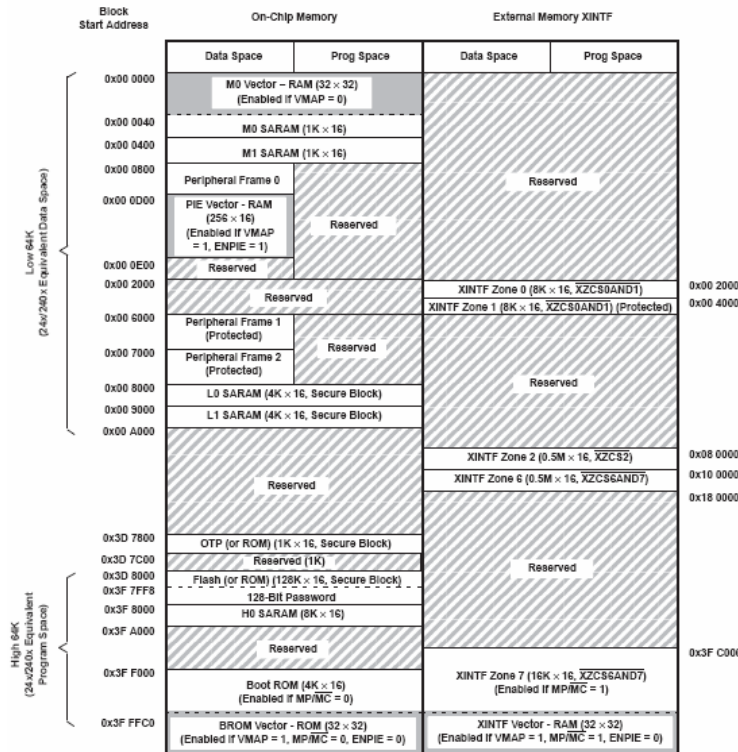


Fig. 3.4. Mappa della memoria [Dat¹]⁽¹⁴⁾

14 Only one of these vector maps—M0 vector, PIE vector, BROM vector, XINTF vector—should be enabled at a time.

NOTES:

A. Memory blocks are not to scale.

B. Reserved locations are reserved for future expansion. Application should not access these areas.

C. Boot ROM and Zone 7 memory maps are active either in on-chip or XINTF zone depending on MP/MC, not in both.

D. Peripheral Frame 0, Peripheral Frame 1, and Peripheral Frame 2 memory maps are restricted to data memory only. User program cannot access these memory maps in program space.

E. "Protected" means the order of Write followed by Read operations is preserved rather than the pipeline order.

F. Certain memory ranges are EALLOW protected against spurious writes after configuration.

G. Zones 0 and 1 and Zones 6 and 7 share the same chip select; hence, these memory blocks have mirrored locations.

3.3. Struttura e routine del software per l'implementazione dell'osservatore

Il software per l'implementazione dell'osservatore è costituito da un insieme di routine che vengono eseguite dal DSP in una sequenza che dipende da vari hardware interrupts. Queste routine sono scritte in linguaggio C/C++ (solo 2 o 3 istruzioni vanno scritte in assembler), inserite nel file DSP281x_Osservatore.c, che è riportato in Appendice. Al fine di accedere alle varie periferiche di controllo, come ad esempio l'Event Manager "EV" (unità di generazione PWM), l'Analog-Digital Converter (ADC) o la Serial Peripheral Interface (SPI), sono utilizzate alcune routine dalla libreria del DSP, [TIN¹, TIN², TIN³]; nel prossimo paragrafo si farà una spiegazione più dettagliata. Queste librerie sono state scaricate gratuitamente da internet. Le definizioni delle varie costanti e variabili utilizzate nel file DSP281x_Osservatore.c sono memorizzate nel file DSP281x_Constant.h. Nel file DSP281x_Device.h si definiscono diverse caratteristiche da impostare nel DSP come ad esempio: il tipo di scheda da utilizzare (DSP28_F2812, DSP28_F2811 o DSP28_F2810), la lunghezza dei dati (int, long, float, ecc.), le definizioni della CPU, ecc. Nel caso del DSP281x_Examples.h si definisce il SYSCLKOUT, cioè la velocità del clock del CPU. Nel DSP281x_AD7568.h si dichiarano le variabili utilizzate per inviare l'informazione al Digital-Analogic Converter "AD7568", e per ultimo si includono le due librerie da utilizzare, qmath.h ed IQmathlib.h. Tutti questi file sono anche elencati nell'appendice.

Ci sono 6 routine nel DSP281x_Osservatore.c: una "void main(void)" routine, una "void init_eva(void)" routine, una "void spi_fifo_init(void)" routine, una "void AD7568_init(void)" routine, una "void AD7568_write(int16 buf[8])" routine ed una "interrupt void motorcontrol(void)" routine. La figura 3.5 mostra un diagramma a blocchi dove si descrive la struttura del sistema del controllo e le principali operazioni che vengono effettuate nelle varie routine.

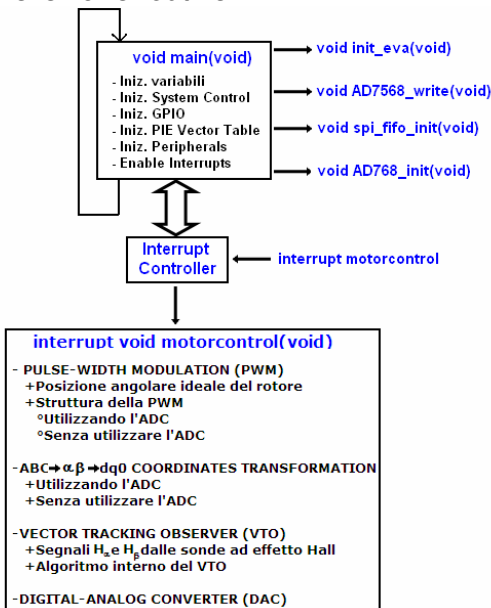


Fig. 3.5. Struttura delle routine eseguite per l'implementazione del sistema di controllo

3.3.1. Periferiche e librerie del DSP (eZdsp™ TMS320F2812) utilizzate per l'implementazione dell'osservatore

Per l'implementazione dell'osservatore sono state utilizzate diverse periferiche disponibile all'interno del DSP (eZdsp™ TMS320F2812), queste periferiche sono elencate e descritte da seguito:

3.3.1.1. Event Manager (EV)

I moduli Event-Manager (EV) forniscono una vasta gamma di funzioni e caratteristiche che sono particolarmente utili nei sistemi di controllo di movimento e sistemi di controllo dei motore. I moduli EV sono composti da General_purpose (GP) timers, full-compare/PWM units, capture units e quadrature-encoder pulse (QEP) circuits. I due moduli EV, EVA ed EVB, sono periferiche identiche, destinati per applicazioni "multi-axis/motion-control".

Ogni EV è in grado di controllare un intero ponte trifase ad IGBT. Ogni EV ha anche altre due PWM con uscite non complementari.

EVA e EVB timers, compare units e capture units funzionano in modo identico. Tuttavia, il timer/unit differiscono nei nomi per l'EVA e l'EVB. La tabella 3.1 mostra le caratteristiche e le funzionalità disponibili per i moduli EV.

Tab. 3.1. Nomi dei moduli e segnali per l'EVA e l'EVB [TIN¹⁵]

Event Manager Modules	EVA		EVB	
	Module	Signal	Module	Signal
GP Timers	GP Timer1	T1PWM/T1CMP	GP Timer3	T3PWM/T3CMP
	GP Timer2	T2PWM/T2CMP	GP Timer4	T4PWM/T4CMP
Compare units	Compare 1	PWM 1/2	Compare 4	PWM 7/8
	Compare 2	PWM 3/4	Compare 5	PWM 9/10
	Compare 3	PWM 5/6	Compare 6	PWM 11/12
Capture units	Capture 1	CAP1	Capture 4	CAP4
	Capture 2	CAP2	Capture 5	CAP5
	Capture 3	CAP3	Capture 6	CAP6
QEP channels	QEP	QEP1	QEP	QEP3
		QEP2		QEP4
		QEP11		QEP12
External timer inputs	Timer-direction external clock	TDIRA	Timer-direction external clock	TDIRB
		TCLKINA		TCLKINB

¹⁵ In the 240x™-compatible mode, the $\overline{T1CTRIP} / \overline{T2CTRIP}$ pin functions as $\overline{PDPINTA}$ and the $\overline{T3CTRIP} / \overline{T4CTRIP}$ pin functions as $\overline{PDPINTB}$. The 240x™ is a trademark of Texas Instruments.

External timer-compare trip inputs	Compare	$\overline{C1TRIP}$		$\overline{C4TRIP}$
		$\overline{C2TRIP}$		$\overline{C5TRIP}$
		$\overline{C3TRIP}$		$\overline{C6TRIP}$
External compare-timer trip inputs		$\overline{T1CTRIP} / \overline{T2CTRIP}$		$\overline{T3CTRIP} / \overline{T4CTRIP}$
External trip inputs		$\overline{PDPINTA}^{(15)}$		$\overline{PDPINTB}^{(15)}$
External ADC SOC trigger inputs		EVASOC		EVBSOC

I moduli EVA e EVB hanno un insieme di registri periferiche identici, i cui indirizzi di memoria per EVA iniziano da 7400h e per EVB iniziano da 7500h. Le seguenti figure mostrano l'architettura dell'Event Manager (EV):

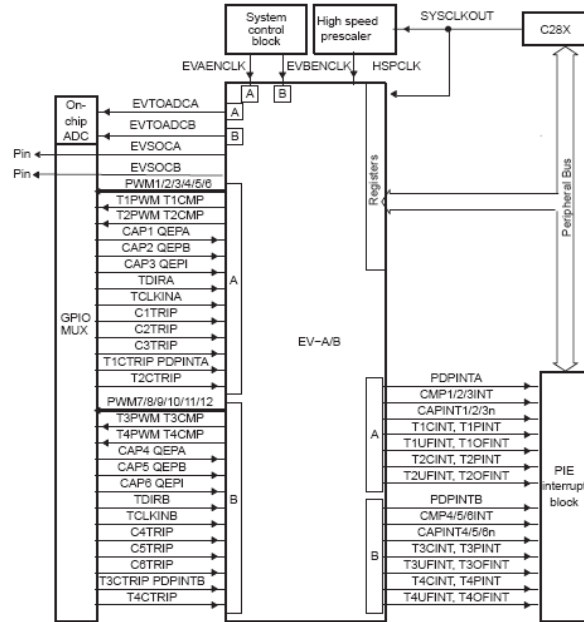


Fig. 3.6. Event Manager (EV) Device Interface [TIN¹] ⁽¹⁶⁾

¹⁶ **Note:** EXTCONA is an added control register to enable and disable the added/modified features. It is required for compatibility with 240x EV. EXTCONA enables and disables the additions and modifications in features. All additions and modifications are disabled by default to keep compatibility with 240x EV. See Section 5.7 for details about the EXTCONx register.

¹⁷ **NOTE A:** The EVB module is similar to the EVA module.

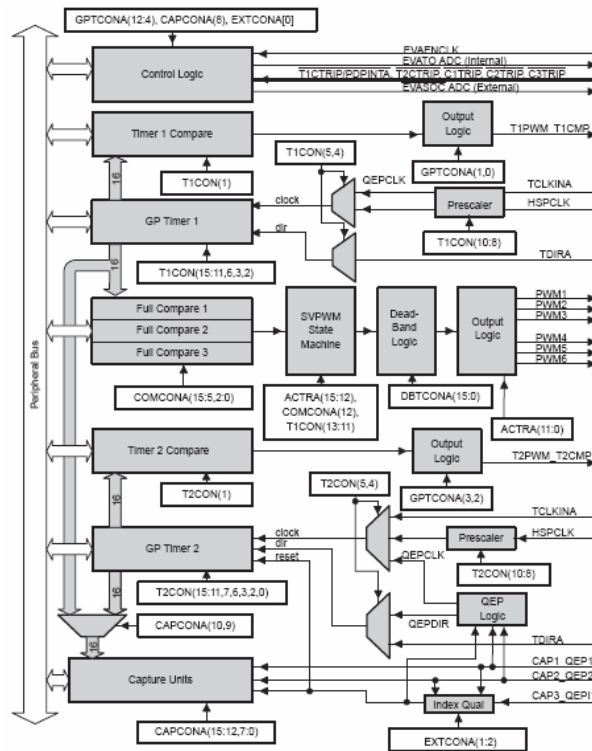


Fig. 3.7. Diagramma a blocchi funzionale del Event Manager A [TIN¹] (17)

L'impostazione di questa periferica per la generazione della PWM sarà descritta nel paragrafo 3.3.7.1.

3.3.1.2. Analog-Digital Converter (ADC)

Il modulo TMS320F28xTM ADC è un Convertitore Analogico-Digitale (ADC) con capacità massima per canale di 12-bit. Il circuito analogico di questo convertitore comprende l'Analog Multiplexers (MUX) di ingresso ed uscita, i circuiti Sample-and-hold (S/H), la conversione di base, regolatori di tensione ed altri circuiti analogici di supporto. I circuiti digitali comprendono un convertitore programmabile di sequenza, registri di risultati, interfaccia per i circuiti analogici, interfaccia con dispositivi attraverso i peripheral bus ed interfaccia con altri moduli.

Il modulo ADC dispone di 16 canali, configurabili come due moduli indipendenti di 8 canali gestiti dall'EVA ed EVB. I due moduli indipendenti a 8 canali possono essere aggiunti in cascata per formare un unico modulo di 16 canali. Sebbene ci sono canali multipli di ingresso e due sequenze, c'è un solo convertitore nel modulo ADC. La figura 3.8 mostra il diagramma a blocchi del modulo ADC.

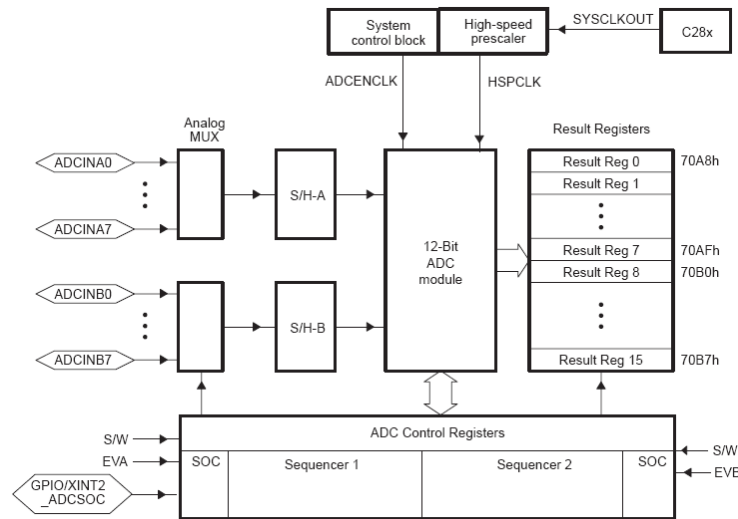


Fig. 3.8. Diagramma a blocchi del Modulo ADC [TIN²]

I due moduli di 8 canali hanno la capacità di fare in auto-sequenza una serie di conversioni, ogni modulo può selezionare uno dei rispettivi otto canali disponibili attraverso un Analog MUX. Nella modalità a cascata, le funzioni di auto-sequenza fanno una unica sequenza a 16 canali. Su ogni sequenza, una volta che la conversione è completata, il valore del canale selezionato è memorizzato nel rispettivo registro ADCRESULT. L'autosequencing permette al sistema di convertire lo stesso canale più volte, permettendo all'utente di eseguire algoritmi con sovra-campionamento. Questo offre una maggiore risoluzione rispetto ai risultati ottenuti con la tradizionale conversione a singolo campione.

Tra le funzioni del modulo ADC si trovano [TIN²]:

- 12-bit ADC core with built-in dual sample-and-hold (S/H);
- Simultaneous sampling or sequential sampling modes;
- Analog input: 0 V to 3 V;
- Fast conversion time runs at 25 MHz, ADC clock, or 12.5 MSPS;
- 16-channel, multiplexed inputs;
- Autosequencing capability provides up to 16 "autoconversions" in a single session. Each conversion can be programmed to select any 1 of 16 input channels;
- Sequencer can be operated as two independent 8-state sequencers or as one large 16-state sequencer (i.e., two cascaded 8-state sequencers);

- Sixteen result registers (individually addressable) to store conversion values
 - The digital value of the input analog voltage is derived by:

$$\text{Digital Value} = 4095 \times \frac{\text{Input Analog Voltage} - \text{ADCL0}}{3}$$
- Multiple triggers as sources for the start-of-conversion (SOC) sequence
 - S/W – Software immediate start;
 - EVA – Event manager A (multiple event sources within EVA);
 - EVB – Event manager B (multiple event sources within EVB);
 - External pin.
- Flexible interrupt control allows interrupt request on every end-of-sequence (EOS) or every other EOS;
- Sequencer can operate in “start/stop” mode, allowing multiple “time-sequenced triggers” to synchronize conversions;
- EVA and EVB triggers can operate independently in dual-sequencer mode
- Sample-and-hold (S/H) acquisition time window has separate prescale control;
- Sequencer override mode enhancement is available only in the F2810/F2811/F2812 silicon after revision B.

3.3.1.3. Serial Peripheral Interface (SPI)

La Serial Peripheral Interface (SPI) è una porta seriale sincrona di input/output (I/O) ad alta velocità che consente ad un flusso di bit di lunghezza programmata (da 1 a 16-bits) che di essere spostato in entrata ed in uscita dal dispositivo ad un programmato rate di trasferimento di bit. La SPI è normalmente utilizzata per le comunicazioni tra il controllore del DSP e le periferiche esterne o di un altro controllore. Le applicazioni tipiche includono I/O esterni o periferiche di espansione tramite dispositivi come registri di spostamento, driver dello schermo, e Convertitori Analogico-Digitale (ADC). Multidevice communications sono supportati dal funzionamento master/slave del SPI.

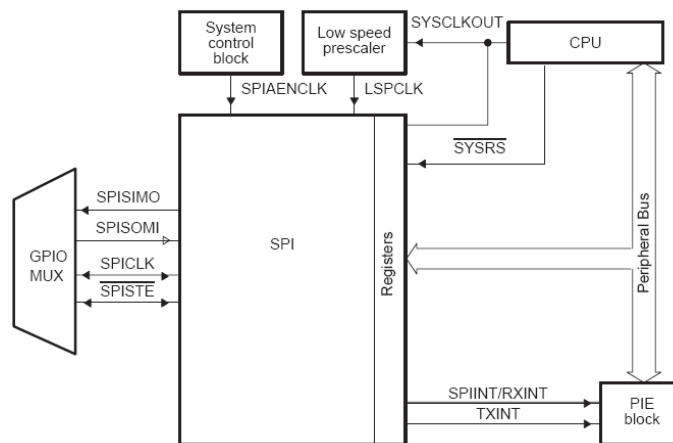


Fig. 3.9. SPI CPU Interface [TIN³]

Tra le funzioni del modulo SPI si trovano [TIN³]:

- Four external pins:
 - SPISOMI: SPI slave-output/master-input pin
 - SPISIMO: SPI slave-input/master-output pin
 - SPISTE: SPI slave transmit-enable pin
 - SPICLK: SPI serial-clock pin

Note: All four pins can be used as GPIO, if the SPI module is not used.

- Four clocking schemes (controlled by clock polarity and clock phase bits) include:
 - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
 - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
 - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.

- Two operational modes: master and slave
- Baud rate: 125 different programmable rates. The maximum baud rate that can be employed is limited by the maximum speed of the I/O buffers used on the SPI pins. See the device-specific data sheet for more details.
- Data word length: one to sixteen data bits

- Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive and transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt-driven or polled algorithms.
- 12 SPI module control registers: Located in control register frame beginning at address 7040h.

Note: All registers in this module are 16-bit registers that are connected to Peripheral Frame 2. When a register is accessed, the register data is in the lower byte (7-0), and the upper byte (15-8) is read as zeros. Writing to the upper byte has no effect.

Il modulo SPI ha due funzioni avanzate:

- FIFO a 16 livelli per trasmissione/ricezione
- Controllo di trasmissione ritardato.

Nel seguente diagramma a blocchi il modulo SPI è in modalità slave, mostrando i blocchi di controllo di base disponibili nel modulo C28xTM SPI [TIN³].

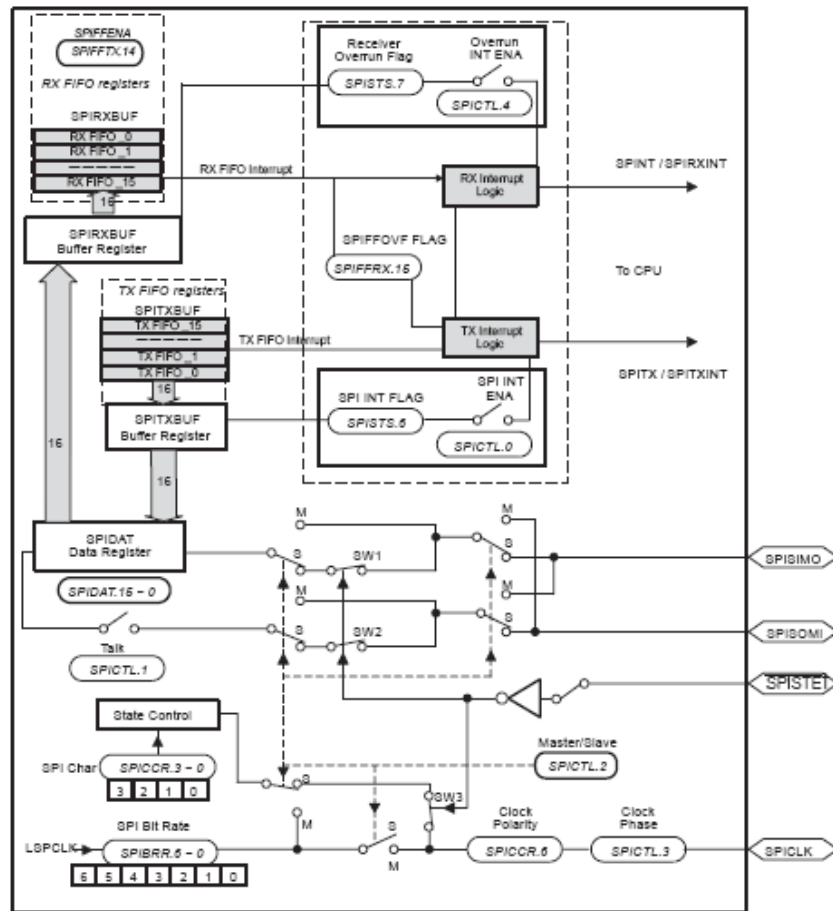


Fig. 3.10. Diagramma a blocchi del modulo C28x™ SPI [TIN³] ⁽¹⁸⁾

¹⁸ SPISITE of a slave device is driven low by the master.

3.3.1.4. Fixed Point Math Library (Qmath Library)

La libreria Fixed Point Math Library (Qmath) [QMA¹] contiene le routine matematiche/trigonometriche che vengono sviluppate in modo che questa sia una libreria facile da usare, e deve essere collegata per l'utente per le diverse applicazioni. La tabella 3.2 riassume l'insieme di funzioni che sono definite in questa libreria, invece la tabella 3.3 indica le prestazioni di ogni funzione della libreria.

Tab. 3.2. FIXED POINT MATH FUNTIONS [QMA¹]

Nome della Funzione	Descrizione
QSIN	Fixed point SIN (Taylor series implementation)
QSINLT	Fixed point SIN (Table look-up and linear interpolation)
QCOS	Fixed point COS (Taylor series implementation)
QCOSLT	Fixed point COS (Table look-up and linear interpolation)
QATAN	Fixed point ATAN (Taylor series implementation)
QSQRT	Fixed point Square-root (Taylor series implementation)
QLOG10	Fixed point LOG10 (Taylor series implementation)
QLOGN	Fixed point LOGN (Taylor series implementation)
QINV1	Reciprocal (32-bit Precision)
QINV2	Reciprocal (16-bit Precision)
QDIV	Division

Tab. 3.3. C28xTM Prestazioni della Libreria QMATH [QMA¹] ⁽¹⁹⁾

Nome della Funzione	Cicli di Esecuzioni	Accuratezza	Memoria di Programma	Formato di Ingresso	Formato di Uscita
QSIN	33 cicli	15 bits	31 parole	Normalizzato Q1.15	Signed Q1.15
QSINLT	25 cicli	14 bits	17 parole	Normalizzato Q1.15	Signed Q1.15
QCOS	33 cicli	15 bits	32 parole	Normalizzato Q1.15	Signed Q1.15
QCOSLT	25 cicli	14 bits	19 parole	Normalizzato Q1.15	Signed Q1.15
QATAN	65 cicli	15 bits	50 parole	Signed Q16.16	Normalizzato Q1.15
QSQRT	45 cicli	14.5 bits	44 parole	Unsigned Q16.16	Unsigned Q8.8
QLOG10	38 cicli	15.5 bits	38 parole	Unsigned Q16.16	Signed Q4.12
QLOGN	38 cicli	15.5 bits	38 parole	Unsigned Q16.16	Signed Q5.11

¹⁹ **Notes:**

1. Execution Cycles mentioned in the table includes the CALL and RETURN (LCR+LRETR).
2. **QCOSLT** and **QSINLT** functions use 256-point look-up table.
3. Branch instruction is totally eliminated in all the above functions.

QINV1	51 cicli	32 bits	15 parole	Signed Q(x)	Signed Q(31-x)
QINV2	35 cicli	16 bits	14 parole	Signed Q(x)	Signed Q(15-x)
QDIV	54 cicli	32 bits	16 parole	Signed Q(x),Q(y)	Signed Q(16+x-y)

3.3.1.5. Virtual Floating Point Engine (IQmath Library)

La libreria IQmath della Texas Instruments TMS320C28x è una libreria di funzioni matematiche altamente ottimizzata e ad alta precisione per programmatori in C/C++ senza soluzione per portare un algoritmo in formato virgola mobile nel codice in formato a punto fisso nei dispositivi TMS320C28x. Queste routine sono generalmente utilizzati per calcoli informatici di applicazioni in tempo reale in cui una ottimale velocità di esecuzione ed alta precisione è necessaria. Con l'utilizzo di tali procedure è possibile raggiungere notevole velocità di esecuzione di codice equivalente scritto in linguaggio standard ANSI C. Inoltre, fornendo le funzioni pronto per l'uso ad alta precisione, la libreria TI IQmath può abbreviare notevolmente lo sviluppo in termini di tempo le applicazioni nel DSP.

Le funzioni incluse nella libreria IQmath sono descritte nella seguente tabella [IQm¹]:

Tab. 3.4. IQMATH FUNCTIONS [IQm¹]

Nome della Funzione	Descrizione
Funzioni Trigonometriche	
IQNsin	High precision SIN (Input in radians)
IQNsinPU	High precision SIN (Input in per-unit)
IQNcos	High precision COS (Input in radians)
IQNcosPU	High precision COS (Input in per-unit)
IQNatan2	4-quadrant ATAN (Output in radians)
IQNatan2PU	4-quadrant ATAN (Output in per-units)
IQNatan	Arctangent (Output in radians)
Funzioni Matematiche	
IQNsqr	High precision square root
IQNisqr	High precision inverse square root
IQNmag	Magnitude Square : $\sqrt{A^2 + B^2}$
Funzioni Aritmetiche	
IQNmpy	IQ Multiplication
IQNrmpy	IQ Multiplication with rounding
IQNrsmPy	IQ Multiplication with rounding & saturation
IQNmpyl32	Multiply IQ with "long" integer
IQNmpyl32int	Multiply IQ with "long", return integer part
IQNmpyl32frac	Multiply IQ with "long", return fraction part
IQNmpylQX	Multiply two 2-different IQ number
IQNdiv	Fixed point division

Conversioni di Formato	
IQN	Converts float to IQ value
IQNtoF	IQ to Floating point
atoIQN	Float ASCII string to IQ
IQNint	Extract integer portion of IQ
IQNfrac	Extract fractional portion of IQ
IQtoIQN	Convert IQ number to IQN number (32-bit)
IQNtoIQ	Convert IQN (32-bit) number to IQ number
IQtoIQN	Convert IQ number to QN number (16-bit)
QNtoIQ	Convert QN (16-bit) number to IQN number
Miscellanea	
IQNsat	Saturate to IQ number
IQNabs	Absolute value of IQ number

La tabella 3.5 indica le prestazioni di ogni funzione della libreria IQmath [IQm¹]:

Tab. 3.5. C28xTM Prestazioni della Libreria IQMATH [IQm¹] ⁽²⁰⁾

Nome della Funzione	IQ Format (N)	Cicli di Esecuzioni	Accuratezza	Memoria di Programma	Formato di Ingresso	Formato di Uscita	Remarks
Funzioni Trigonometriche							
IQNsin	1 to 29	46	30 bits	49 words	IQN	IQN	
IQNsinPU	1 to 29	40	30 bits	41 words	IQN	IQN	
IQNcos	1 to 29	44	30 bits	47 words	IQN	IQN	
IQNcosPU	1 to 29	38	29 bits	39 words	IQN	IQN	
IQNatan2	1 to 29	109	26 bits	123 words	IQN	IQN	
IQNatan2PU	1 to 29	117	27 bits	136 words	IQN	IQN	
IQNatan	1 to 29	109	25 bits	123 words	IQN	IQN	
Funzioni Matematiche							
IQNsqrt	1 to 30	63	29 bits	66 words	IQN	IQN	
IQNisqrt	1 to 30	64	29 bits	69 words	IQN	IQN	
IQNmag	1 to 30	86	29 bits	96 words	IQN	IQN	

²⁰ **Notes:**

- Execution cycles & Program memory usage mentioned in the Table assumes IQ24 format.
 Ø Execution cycles may vary by few cycles for some other IQ format.
 Ø Program memory may vary by few words for some other IQ format.
- Execution Cycles mentioned in the table includes the CALL and RETURN (LCR + LRETR) and it assumes that the IQmath table is loaded in internal memory.

Funzioni Aritmetiche							
IQNmpy	1 to 30	~ 6 cicli	32 bits	N/A	IQN*IQN	IQN	INTRINSI C
IQNrmpy	1 to 30	17	32 bits	13 words	IQN*IQN	IQN	
IQNrsmPy	1 to 30	21	32 bits	21 words	IQN*IQN	IQN	
IQNmpyl32	1 to 30	~ 4 cicli	32 bits	N/A	IQN*long	IQN	C-MACRO
IQNmpyl32int	1 to 30	22	32 bits	16 words	IQN*long	long	
IQNmpyl32frac	1 to 30	24	32 bits	20 words	IQN*long	IQN	
IQNmpylQX	1 to 30	~ 7 cicli	32 bits	N/A	IQN1*IQN2	IQN	INTRINSI C
IQNdiv	1 to 30	63	28 bits	71 words	IQN/IQN	IQN	
Conversioni di Formato							
IQN	1 to 30	N/A	N/A	N/A	Float	IQN	C-MACRO
IQNtoF	1 to 30	22	N/A	20 words	IQN	Float	
atoIQN	1 to 30	N/A	N/A	143 words	char	IQN	
IQNint	1 to 30	14	32 bits	8	IQN	long	
IQNfrac	1 to 30	17	32 bits	12	IQN	IQN	
IQtoIQN	1 to 30	~ 4 cicli	N/A	N/A	GLOBAL_ Q	IQN	C-MACRO
IQNtoIQ	1 to 30	~ 4 cicli	N/A	N/A	IQN	GLOBAL_ Q	C-MACRO
IQtoQN	1 to 30	~ 4 cicli	N/A	N/A	GLOBAL_ Q	QN	C-MACRO
QNtoIQ	1 to 30	~ 4 cicli	N/A	N/A	QN	GLOBAL_ Q	C-MACRO
Miscellanei							
IQNsats	1 to 30	~ 7 cicli	N/A	N/A	IQN	IQN	INTRINSI C
IQNabs	1 to 30	~ 2 cicli	N/A	N/A	IQN	IQN	INTRINSI C

3.3.2. Routine di inizio (void main(void))

La routine di inizio viene eseguita una sola volta, prima della prima chiamata delle principali routine, ed è utilizzata per scopi di inizializzazione. Una delle principali operazioni che vengono eseguite è l'inizializzazione delle variabili utilizzate nella routine principale (interrupt void motorcontrol(void)):

```
...
/* Initialize variables */
posiz1 = 0x0000;
posiz2 = 0x0000;
posiz3 = 0x0000;
posiz_prov = 0x0000;
il = 0x0000;
acc_obs_error = 0x0000;
omega_hat = 0x0000;
theta_hall = 0x0000;
...
```

Dopo questo si iniziano una serie di passi, i primi due sono: Inizializzazione del System Control, dove si inizializza il PLL, si accende il Peripheral Clock e si imposta il Clock Pre-scaler. Il watchdog è disabilitato. Successivamente c'è l'inizializzazione del GPIO (General-Purpose Input/Output), dove si possono impostare i pins del GPIO o come ingresso o uscita, a seconda di quello che sia richiesto.

```
...
/* Step 1. Initialize System Control:
PLL, WatchDog, enable Peripheral Clocks
This function is found in the DSP281x_SysCtrl.c file.*/
InitSysCtrl();

/* Step 2. Initialize GPIO:
This function is found in the DSP281x_Gpio.c file and
illustrates how to set the GPIO to it's default state.
InitGpio(); Skipped for this program */
...
```

Tutti i numeri sono scritti in formato esadecimale. Per questo programma si imposta il HSPCLK (High-Speed Clock) a 25MHz, e poi si abilitano i pins per la PWM, i pins del GPIO, le funzionalità del SPI e le uscite DAC:

```
...
/* For this program, set HSPCLK to SYSCLKOUT / 6
(25MHz assuming 150MHz SYSCLKOUT) */
EALLOW;
SysCtrlRegs.HISPCP.all = 0x0003;          // HSPCLK = SYSCLKOUT/6
EDIS;

/* Initialize only GPAMUX, GPBMUX, GPBDIR and GPFMUX for this
program */
EALLOW;
// Enable PWM pins
GpioMuxRegs.GPAMUX.all = 0x00FF;          // EVA PWM 1-6 pins
// Enable GPIO pins
GpioMuxRegs.GPBMUX.bit.PWM7_GPIOB0 = 0; // Porta Digitale 1 (PD1)
attiva
GpioMuxRegs.GPBMUX.bit.PWM8_GPIOB1 = 0; // Porta Digitale 2 (PD2)
attiva
```

```

    GpioMuxRegs.GPBMUX.bit.PWM9_GPIOB2 = 0; // Porta Digitale 3 (PD3)
attiva
    GpioMuxRegs.GPBMUX.bit.PWM10_GPIOB3 = 0; // Porta Digitale 4 (PD4)
attiva
    GpioMuxRegs.GPBMUX.bit.PWM11_GPIOB4 = 0; // Porta Digitale 5 (PD5)
attiva
    GpioMuxRegs.GPBMUX.bit.PWM12_GPIOB5 = 0; // Porta Digitale 6 (PD6)
attiva
    GpioMuxRegs.GPBDIR.bit.GPIOB0 = 1;          // Impostazione PD1 in
uscita
    GpioMuxRegs.GPBDIR.bit.GPIOB1 = 1;          // Impostazione PD2 in
uscita
    GpioMuxRegs.GPBDIR.bit.GPIOB2 = 1;          // Impostazione PD3 in
uscita
    GpioMuxRegs.GPBDIR.bit.GPIOB3 = 1;          // Impostazione PD4 in
uscita
    GpioMuxRegs.GPBDIR.bit.GPIOB4 = 1;          // Impostazione PD5 in
uscita
    GpioMuxRegs.GPBDIR.bit.GPIOB5 = 1;          // Impostazione PD6 in
uscita
    // Enable SPI functionality
    GpioMuxRegs.GPFMUX.all=0x000D;              // Select GPIOs to be SPI
                                                pins          // except for
                                                SPISOMI Port F MUX
                                                // x000 0000 0000 1101

    // Enable Digital outputs
    GpioMuxRegs.GPFDIR.bit.GPIOF1 = 1; // SPISOMI is used as digital
output
    GpioMuxRegs.GPFDIR.bit.GPIOF6 = 1; // CANTX is used as digital
output
    EDIS;
...

```

Nel passo 3 si azzerano gli interrupts e si inizializza la PIE vector table:

```

...
/* Step 3. Clear all interrupts and initialize PIE vector table:
Disable CPU interrupts */
DINT;

/* Initialize PIE control registers to their default state.
The default state is all PIE interrupts disabled and flags
are cleared.
This function is found in the DSP281x_PieCtrl.c file. */
InitPieCtrl();

/* Disable CPU interrupts and clear all CPU interrupt flags: */
IER = 0x0000;
IFR = 0x0000;

/* Initialize the PIE vector table with pointers to the shell
Interrupt
Service Routines (ISR).
This will populate the entire table, even if the interrupt
is not used in this program. This is useful for debug purposes.
The shell ISR routines are found in DSP281x_DefaultIsr.c.
This function is found in DSP281x_PieVect.c. */
InitPieVectTable();

/* Interrupts that are used here are re-mapped to ISR functions
found
within this file. */

```

```
EALLOW; // This is needed to write to EALLOW protected register
PieVectTable.T1PINT = &motorcontrol;
EDIS; // This is needed to disable write to EALLOW protected
registers
...
```

Nel passo 4 si fa l'inizializzazione di tutti i dispositivi periferici:

```
...
/* Step 4. Initialize all the Device Peripherals:
This function is found in DSP281x_InitPeripherals.c
InitPeripherals(); Not required for this program */
init_eva();
InitAdc();
spi_fifo_init(); // Initialize the SPI
AD7568_init();
...
```

Nell'ultimo passo si abilita l'interrupt dell'EVA e si impostano le sequenze per ADC

```
...
/* Step 5. User specific code, enable interrupts: */

/* Enable EVA */
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE block
PieCtrlRegs.PIEIER2.bit.INTx4 = 1; // PIE, INT2 Group
                                   // Enable CPU Interrupt
                                   // Register
IER |= M_INT2;
2 EINT; // Enable Global
      // Interrupt INTM
      // Enable Global
      // Realtime
      // Interrupt DBGm
      // GP timer 1 period
      // interrupt // EVA
      // Interrupt Mask Register
      // A

EvaRegs.EVAIMRA.bit.T1PINT = 1;

/* Configure ADC */
AdcRegs.ADCMAXCONV.all = 0x0005; // Setup 4 conv's on
SEQ1 and
                                   // 2 conv's on SEQ2
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0000; // Setup conv from ADCINA0
& // ADCINB0
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x0001; // Setup conv from
ADCINA1 &
                                   // ADCINB1
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x0002; // Setup conv from
ADCINA2 &
                                   // ADCINB2
AdcRegs.ADCTRL2.bit.EVA_SOC_SEQ1 = 1; // Enable EVASOC to
start SEQ1
AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 0; // Enable SEQ1 interrupt
// (every EOS)

/* Just sit and loop forever */
for(;;);
...
```

3.3.3. Routine di inizializzazione del EVA (void init_eva(void))

Nell'Event Manager A (EVA) si possono abilitare il Timer1 ed il Timer2, con cui si possono ottenere le uscite PWM1-PWM6. Inoltre, si fanno due prescaler del Timer1 period register, cioè dal valore predefinito 75MHz si fa un prescaler a 25MHz perché è la frequenza massima permessa per l'ADC e poi il secondo prescaler si fa per ottenere un frequenza per il Timer1 period di 15kHz (frequenza di campionamento), ottenendosi come risultato:

$$(3.1) \quad \frac{75\text{MHz}}{15\text{kHz}} \cdot \frac{1}{3} = \frac{5000}{3} = 1666,6667 \Big|_{\text{decimale}} \Rightarrow 0x0682 \Big|_{\text{Hexadecimale}}$$

All'interno della routine dell'EVA si può anche abilitare l'EVASOC (EVA Star of conversion) utilizzato per il Convertitore Analogico-Digitale (ADC).

```
...
/* EVA Configure T1PWM, T2PWM, PWM1-PWM6 */
/* Initialize the timers */

/* Initialize EVA Timer1 */
EvaRegs.T1PR = 0x0682; // Setup Timer1 period register, should be
                        // 0x1388 // (75MHz/15kHz) but it's setup
                        // as 0x1388/3 // because ADConverter
                        // works at 25 MHz
EvaRegs.T1CMPR = 0x3C00; // Setup Timer1 compare value
EvaRegs.T1CNT = 0x0000; // Timer1 counter

//TMODE = continuous up/down
//Timer enable
//Timer compare enable

EvaRegs.GPTCONA.bit.T1TOADC = 1; // Enable EVASOC in EVA
EvaRegs.T1CON.all = 0x1042; // Enable Timer 1 Compare (upcount
mode)

/* Setup T1PWM and T2PWM */
/* Drive T1/T2 PWM by compare logic */
EvaRegs.GPTCONA.bit.TCMPOE = 1;
/* Polarity of GP Timer 1 Compare = Active low */
EvaRegs.GPTCONA.bit.T1PIN = 1;

/* Enable compare for PWM1-PWM6 */
EvaRegs.CMPR1 = 0x0C00;
EvaRegs.CMPR2 = 0x3C00;
EvaRegs.CMPR3 = 0xFC00;
/* Compare action control. Action that takes place on a compare
event */
//output pin 1 CMPR1 - active high
//output pin 2 CMPR1 - active low
//output pin 3 CMPR2 - active high
//output pin 4 CMPR2 - active low
//output pin 5 CMPR3 - active high
//output pin 6 CMPR3 - active low

EvaRegs.ACTRA.all = 0x0666;
EvaRegs.DBTCONA.all = 0x0000; // Disable deadband
EvaRegs.COMCONA.all = 0xA600;
...
```

3.3.4. Routine di inizializzazione dei registri della SPI FIFO (void spi_fifo_init(void))

In questa routine si inizializzano i registri della Serial Peripheral Interface (SPI) per permettere la trasmissione dei dati che si trovano all'interno del programma (in formato digitale) al Convertitore Digitale-Analogico AD7568 [Ana¹]; l'algoritmo usato si riporta da seguito:

```
...
// Initialize SPI FIFO registers

SpiaRegs.SPICCR.bit.SPISWRESET=0; // Reset SPI
SpiaRegs.SPICCR.all=0x000F;      // 16-bit character, data is output
                                // on // rising edge
SpiaRegs.SPICCTL.all=0x0017;     // Interrupt enabled, Master/Slave
                                // XMIT // enabled

SpiaRegs.SPISTS.all=0x0000;
SpiaRegs.SPIBRR=0x0003;          // Baud rate
SpiaRegs.SPIFFTX.all=0xC028;     // Enable FIFO's, set TX FIFO level
to 8
SpiaRegs.SPIFFRX.all=0x0000;     // Disable RX FIFO Interrupt
                                // because // SPISOMI is used as
                                // GPIO pin (digital // output)

SpiaRegs.SPIFFCT.all=0x01;
SpiaRegs.SPIPRI.all=0x0010;
SpiaRegs.SPICCR.bit.SPISWRESET=1; // Enable SPI
SpiaRegs.SPIFFTX.bit.TXFIFO=1;
SpiaRegs.SPIFFRX.bit.RXFIFORESET=1;
...

```

3.3.5. Routine degli indirizzi delle variabili inviate al DAC “AD7568” (void AD7568_init(void))

Questa routine è stata creata con il proposito di indirizzare le variabili da inviare al DAC (AD7568) per la conversione, si tengono a disposizione 8 canali ed ogni canale possiede il suo proprio indirizzo; l'algoritmo creato è riportato nel:

```
...
Ad7568ChAddr[0]=0; // DAC channel 1 address
Ad7568ChAddr[1]=1; // DAC channel 2 address
Ad7568ChAddr[2]=2; // DAC channel 3 address
Ad7568ChAddr[3]=3; // DAC channel 4 address
Ad7568ChAddr[4]=4; // DAC channel 5 address
Ad7568ChAddr[5]=5; // DAC channel 6 address
Ad7568ChAddr[6]=6; // DAC channel 7 address
Ad7568ChAddr[7]=7; // DAC channel 8 address
...

```

3.3.6. Routine per l'invio dei dati al DAC “AD7568” (void AD7568_write(int16 buf[8]))

Questa routine è stata creata per inviare al DAC “AD7568” le variabili del VTO che devono essere visualizzate nell'oscilloscopio per verificare il corretto funzionamento. I segnali da inviare al DAC sono in formato esadecimale a 16-bit, secondo il seguente formato:

- i 12-bits più significativi contengono l'informazione del segnale che si vuole convertire; per questo motivo si utilizza l'operatore logico (&) per togliere i meno significativi.
- i 4-bits meno significativi contengono l'indirizzo corrispondente ad ogni variabile del VTO che si vuole convertire ed anche individua il canale del ADC che sarà utilizzato per la conversione.

Ad ogni segnale dopo queste operazioni si somma 0x8000 con l'intenzione di portarlo dal formato con segno(int) al formato senza segno(Uint), dopodichè esso è trasmesso tramite il Serial Transmit Buffer Register (SPITXBUF) ai canali di ingresso del DAC. Per capire questo, si fa un piccolo esempio:

$$\begin{aligned}
 DAC_buffer[0] &= 0x4444 \rightarrow \text{si memorizza in } buf[0] \\
 &\rightarrow send_buf[0] = 0x4444 \& 0xFFF0 = 0x4440 \\
 &\quad \rightarrow Ad7568ChAddr[0] = 0 \quad (0x0000) \\
 &\rightarrow send_buf[0] = Ad7568ChAddr[0] \mid 0x4440 = 0x4440 \\
 &\quad \rightarrow send_buf[0] = 0x8000 + 0x4440 = 0xC440
 \end{aligned}$$

Quindi, il segnale trasmesso al primo canale del ADC (Canale 1) sarà 0xC440, il quale corrisponde alla variabile memorizzata in DAC_buffer[0] (0x4444) inviata dall'algoritmo del VTO.

```

...
    Uint16 i;
    Uint16 send_buf[8];

    for(i=0;i<8;i++)
    {
        send_buf[i] = buf[i]&0xFFF0;           // remove LSB
nibble      send_buf[i] = Ad7568ChAddr[i] | send_buf[i]; // add DAC
address     send_buf[i] = 0x8000 + send_buf[i]; /* convert from signed to
                                                unsigned integer*/
    }

    for(i=0;i<8;i++)
    {
        SpiaRegs.SPITXBUF=send_buf[i];       // Send data
    }
...

```

All'interno della routine si fanno due contatori utilizzando due cicli "for" con valore iniziale in 0 e finale in 7 (cioè 8 iterazioni). Il primo contatore identifica e converte le variabili da trasmettere, il secondo invece trasmette attraverso la SPITXBUF le variabili ad ogni canale del DAC (AD7568).

3.3.7. Routine principale di controllo del motore (interrupt void motorcontrol(void))

La routine di controllo del motore è la routine principale del software con cui si fa l'implementazione dell'osservatore ad inseguimento vettoriale (ed anche il principale

obiettivo di questa tesi). All'interno di questa routine si trovano le seguenti 4 applicazioni, sviluppate in questa tesi:

- Pulse-Width Modulation PWM (Modulazione della ampiezza d'impulso);
- Trasformazione di coordinate per le Correnti di fase (ABC) → Corrente nel sistema di riferimento fisso ($\alpha\beta$) → Corrente nel sistema di riferimento arbitrario (qd0);
- VECTOR TRACKING OBSERVER VTO (Osservatore ad inseguimento vettoriale);
- Digital-Analog Converter DAC (Convertitore Digitale-Analogico).

La routine principale sarà poi completata con altre applicazioni che esulano dagli scopi di questa tesi.

3.3.7.1. Pulse-Width Modulation PWM (Modulazione della ampiezza d'impulso)

Il segnale PWM [Ele¹] è generato per mezzo di un circuito comparatore mostrato in figura 3.11, in cui vengono confrontate un'onda triangolare u_T (Timer1 period) ad alta frequenza ed un segnale modulante sinusoidale u_S (Compare) di frequenza uguale alla frequenza desiderata in uscita dell'inverter di potenza trifase. Il segnale uscente da tale circuito dipende dal valore relativo delle onde poste a confronto: ad esempio per il semiperiodo positivo (figura 3.11.b) si avrà un'uscita "alta" negli istanti in cui la sinusoide è maggiore della triangolare e "bassa" negli altri.

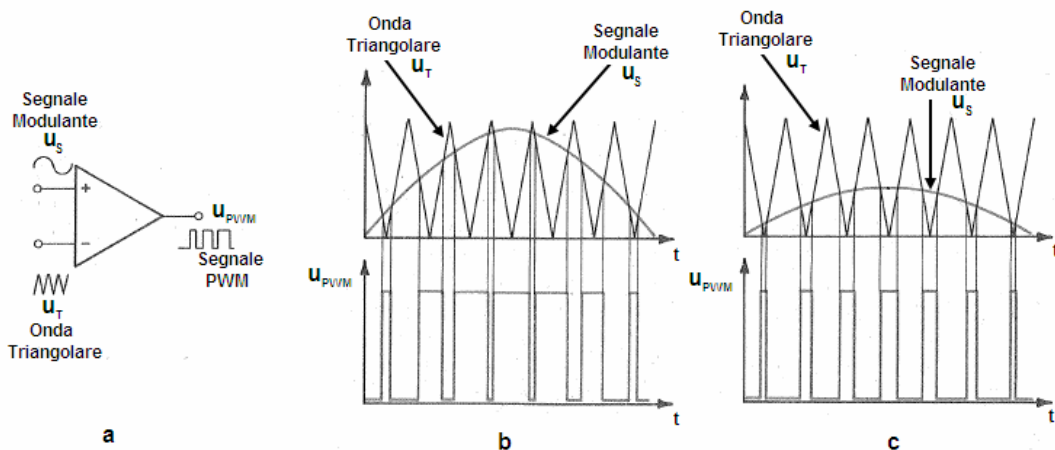


Fig. 3.11. Generazione della PWM (Modulazione ad ampiezza d'impulsi) [Ele¹]

a-. Circuito Comparatore per la generazione della PWM

b,c -. Confronto tra l'onda triangolare u_T e l'onda modulante u_S con diversi valore di ampiezza

Questo segnale viene poi usato per accendere o spegnere gli switch opportuni in modo che anche la forma d'onda di tensione prodotta dall'inverter segua questo andamento.

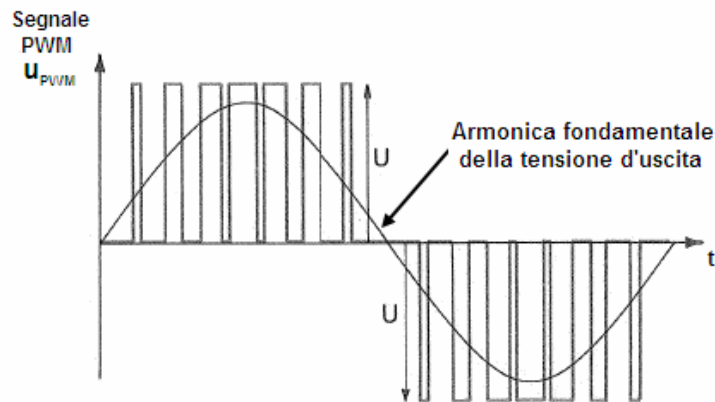


Fig. 3.12. Regolazione della tensione in uscita da un inverter attraverso la modulazione ad ampiezza d'impulsi (PWM) [Ele¹]

Sono stati realizzati due esempi di routine di modulazione: il primo utilizzando anche il Convertitore Analogico-Digitale ed il secondo a frequenza costante.

Nel primo caso, il fatto di utilizzare l'ADC serve per verificare che detto convertitore fa la conversione del segnale analogico proveniente dal potenziometro (compresso tra 0 e +5V) al rispettivo valore in digitale tra 0x0000 e 0x0FFF (il convertitore lavora a 12 bits); allora questo segnale esterno fa variare l'ampiezza A del segnale modulante in modo che si abbia un duty cycle che varia tra 0 e 1, centrato in 0.5. La rapidità con la quale varia il suo duty cycle dipenderà direttamente dalla frequenza del segnale modulante. Le figure 3.13 e 3.14 mostrano questo comportamento:

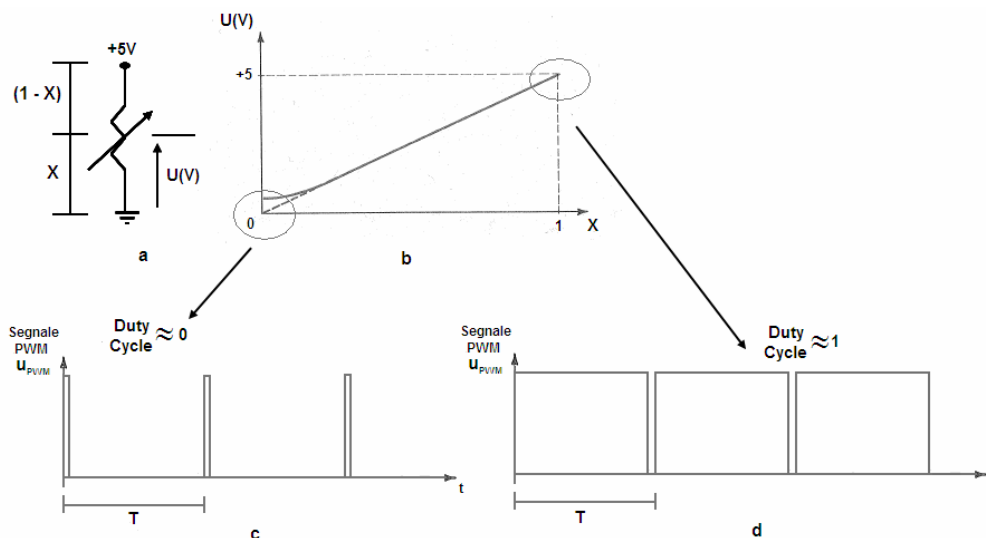


Fig. 3.13. Variazione dell'ampiezza della modulante con azione diretta sul valore del Duty Cycle

a-. Potenziometro alimentato tra 0 e +5V

b-. Curva di variazione del Potenziometro di tensione rispetto alla sua posizione

c-. Segnale PWM con Duty Cycle circa 0

d-. Segnale PWM con Duty Cycle circa 1

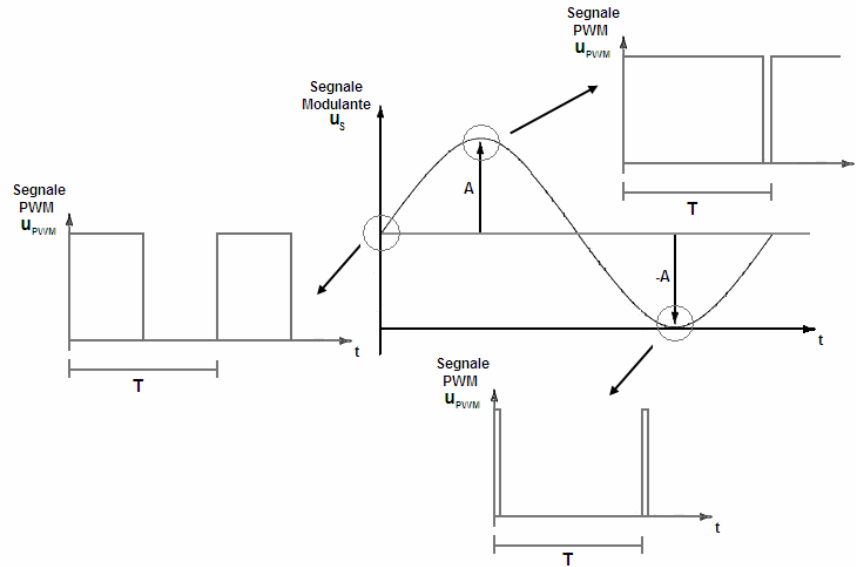


Fig. 3.14. Variazione del Duty Cycle proporzionale alla ampiezza e frequenza del segnale modulante u_s

In questa applicazione si genera l'onda triangolare u_T ad una frequenza di 15kHz: per raggiungere questo abbiamo impostato il timer1 period al valore in esadecimale 0x0682 (come descritto nel paragrafo 3.3.3), simultaneamente si genera un segnale di posizione angolare ideale per le 3 fasi (a 0° , $+120^\circ$ e -120°) i quali sono gli argomenti delle funzioni sinusoidale con lo scopo di produrre i segnali modulanti u_s per ogni fase con una ampiezza pari a 0x0682 ed una frequenza pari a

$$\left(\omega = 2 \cdot \pi \cdot f = 400 \frac{\text{rad}}{\text{s}} \Rightarrow f = \frac{400 \frac{\text{rad}}{\text{s}}}{2 \cdot \pi} = 63,661977 \text{Hz} \right). \text{ Il risultato sar\`a un segnale}$$

PWM con le stesse caratteristiche di ampiezza e frequenza del segnale modulante. L'algoritmo con cui si genera la posizione angolare di riferimento delle tre fase (a 0° , $+120^\circ$ e -120°) \u00e8 mostrato nel seguito:

```
...
/*##### PULSE-WIDTH MODULATION (PWM)
#####*/

/* Posizione angolare principale (Riferimento) */

    vts_int16 = Dtheta_hex3;
/* Si dichiara in formato Q.15(int16), per poterla visualizzare
attraverso il Digital-Analogic Converter (DAC) */

    posiz1 = posiz1 + vts_int16;
/* Posizione angolare ideale del rotore del motore, utilizzata come
riferimento */

    posiz2 = posiz1 + _2div3_;
```

```
/* Posizione angolare ideale del rotore del motore spostata in
+120gradi elettrici */

    posiz3 = posiz1 - _2div3_;
/* Posizione angolare ideale del rotore del motore spostata in -
120gradi elettrici */
...

```

La generazione dei segnali modulante per ogni fase è stata effettuata utilizzando la libreria Fixed Point Math Library (Qmath) [QMA¹]:

```
...
    Sin1 = qsin(posiz1);
    Sin2 = qsin(posiz2);
    Sin3 = qsin(posiz3);
/* PWM Sinusoidale */
...

```

L'algoritmo utilizzato nel primo caso in cui si utilizza il Convertitore Analogico-Digitale, è mostrato da seguito:

```
...
/***** Utilizzando l'ADC *****/

    MisuraestPWM1 = AdcRegs.ADCRESULT0 >> 4;
    MisuraestPWM2 = AdcRegs.ADCRESULT1 >> 4;
    MisuraestPWM3 = AdcRegs.ADCRESULT2 >> 4;
/* Acquisizione analogica (Potenziometro) per la conversione a
digitale, si tolgono i primi 4 bits per condizioni del ADC. Con
questo potenziometro riesco ad ottenere un duty cycle variabile, cioè
da libera impostazione */

    prod1 = Sin1 * MisuraestPWM1;
    prod2 = Sin2 * MisuraestPWM2;
    prod3 = Sin3 * MisuraestPWM3;
/* Prodotto tra il valore Misurato per il ADC ed il seno
(l'argomento del seno sarà la posizione angolare ideale di
riferimento) */

    prod1_shiftato = prod1 >> 12;
    prod2_shiftato = prod2 >> 12;
    prod3_shiftato = prod3 >> 12;
/* Si tolgono i 12 bits meno significativi per portarlo al formato
1.15 */

    p1 = prod1_shiftato + 0x8000;
    p2 = prod2_shiftato + 0x8000;
    p3 = prod3_shiftato + 0x8000;
/* Si porta al formato senza segno hexadecimal 16.0 */

    q1 = p1 * (0x0682);
    q2 = p2 * (0x0682);
    q3 = p3 * (0x0682);
/* Si prescala per ottenere come ampiezza massima della modulante
(sinusoidale) il valore in hexadecimal di 0x0682 */

    r1 = q1 / (0xFFFF);
    r2 = q2 / (0xFFFF);
    r3 = q3 / (0xFFFF);

```

```

/* Si fa un shift di 16 bits alla sinistra per portarlo al formato
0.16 */

EvaRegs.CMPR1 = r1;
EvaRegs.CMPR2 = r2;
EvaRegs.CMPR3 = r3;
/* Uscita della PWM Sinusoidale alla frequenza desiderata
f=(400rad/s)/(2*pigreco)=63,66197 Hz */
...

```

Nel secondo caso, non si utilizza l'ADC e quindi si ha che l'ampiezza del segnale modulante è fissa ad un valore massimo in esadecimale pari a 0x0682 ed una

frequenza pari a $\left(\omega = 2 \cdot \pi \cdot f = 400 \frac{\text{rad}}{\text{s}} \Rightarrow f = \frac{400 \frac{\text{rad}}{\text{s}}}{2 \cdot \pi} = 63,661977 \text{Hz} \right)$, e poi si

aggiunge l'offset pari alla metà del valore massimo della modulante, cioè 0x0341. L'algoritmo eseguito in questo secondo caso si mostra da seguito:

```

...
/***** Senza utilizzare l'ADC *****/

e1 = Sin1 * 0x0682;
e2 = Sin2 * 0x0682;
e3 = Sin3 * 0x0682;
/* Il prodotto mi prescala l'onda per ottenere come ampiezza massima
della modulante (sinusoidale) il valore in hexadecimale di 0x0682 */

e1_shiftato = e1 >> 16;
e2_shiftato = e2 >> 16;
e3_shiftato = e3 >> 16;
/* Si prende i 16 bits più significativi dal prodotto precedente */

g1 = e1_shiftato;
g2 = e2_shiftato;
g3 = e3_shiftato;
/* Si riporta in formato Q.15(int16) per poter visualizzare il
segnale attraverso il DAC */

g1_1 = g1 + 0x0341;
g2_2 = g2 + 0x0341;
g3_3 = g3 + 0x0341;
/* Si aggiunge l'offset, pari alla ampiezza massima della modulante
(0x0682) diviso 2 */

EvaRegs.CMPR1 = g1_1;
EvaRegs.CMPR2 = g2_2;
EvaRegs.CMPR3 = g3_3;
/* Uscita della PWM Sinusoidale alla frequenza desiderata
f=(400rad/s)/(2*pigreco)=63,66197 Hz */
...

```

3.3.7.2. Trasformazione di coordinate per le Correnti di fase (ABC) → Corrente nel sistema di riferimento fisso ($\alpha\beta$) → Corrente nel sistema di riferimento arbitrario (qd0)

Questa seconda applicazione si è creata per poter fare il controllo di corrente di fase di ingresso al motore (controllo che non fa parte di questa tesi, ma per posteriori lavori è importante), quindi per questa applicazione si ha anche due possibilità: la prima utilizzando il Convertitore Analogico-Digitale per leggere il segnale di corrente inviato dal trasduttore di corrente e la seconda supponendo che abbiamo correnti di fase a valori fissi.

La trasformazione delle correnti di fase (ABC) al sistema di riferimento fisso ($\alpha\beta$), si fa attraverso la trasformata di Clark come di seguito riportato:

$$(3.2) \quad I_{abc} = \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}; I_{\alpha\beta} = \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} \Rightarrow \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}$$

Per un sistema bilanciato si ha che la somma delle correnti è pari a zero, cioè $I_a + I_b + I_c = 0$, quindi si ottiene che:

$$(3.3) \quad \begin{cases} I_a + I_b + I_c = 0 \Rightarrow I_c = -I_a - I_b \\ I_\alpha = \frac{2}{3} \cdot I_a - \frac{1}{3} \cdot I_b - \frac{1}{3} \cdot [-I_a - I_b] = \frac{2}{3} \cdot I_a - \frac{1}{3} I_b + \frac{1}{3} \cdot I_a + \frac{1}{3} \cdot I_b \Rightarrow I_\alpha = I_a \\ I_\beta = \frac{1}{\sqrt{3}} \cdot I_b - \frac{1}{\sqrt{3}} \cdot I_c \end{cases}$$

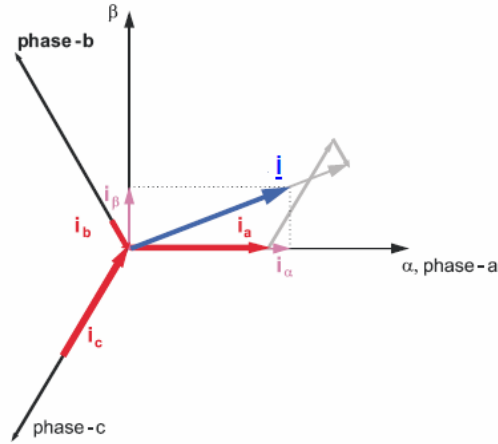


Fig. 3.15. Trasformata di Clark [Fre¹]

Per trasformare ora le correnti del sistema di riferimento fisso ($\alpha\beta$) al sistema di riferimento arbitrario (qd) si utilizza la trasformata di Park, è importante chiarire che l'angolo θ si misura rispetto all'asse q e “non rispetto all'asse d”, per cui si ricava che:

$$(3.4) \quad \begin{cases} I_{qd} = I_q - jI_d \\ I_{\alpha\beta} = \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix}; I_{qd0} = \begin{bmatrix} I_d \\ I_q \end{bmatrix} \Rightarrow \begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} \sin(\theta) & -\cos(\theta) \\ \cos(\theta) & \sin(\theta) \end{bmatrix} \cdot \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} \end{cases}$$

Quindi le correnti nel sistema di riferimento arbitrario (qd) sono:

$$(3.5) \quad \begin{cases} I_d = \sin(\theta) \cdot I_\alpha - \cos(\theta) \cdot I_\beta \\ I_q = \cos(\theta) \cdot I_\alpha + \sin(\theta) \cdot I_\beta \end{cases}$$

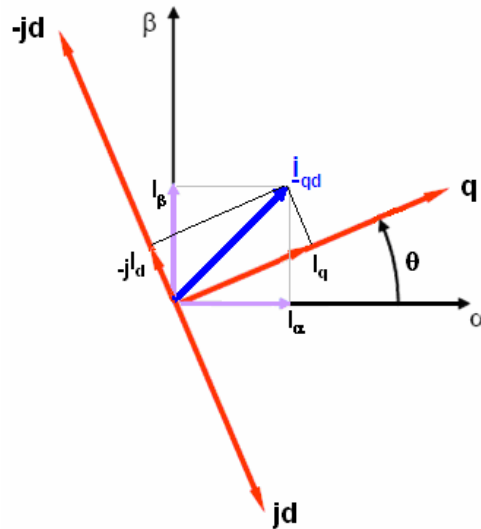


Fig. 3.16. Trasformata di Park [Fre¹]

L'algoritmo quando si utilizza il Convertitore Analogico-Digitale si presenta come nel seguito:

```
...
/*##### ABC → ALPHA,BETA → DQ COORDINATES TRANSFORMATION
#####*/

/***** Utilizzando l'ADC *****/

Misura_Ia = AdcRegs.ADCRESULT3 >> 4;
Misura_Ib = AdcRegs.ADCRESULT4 >> 4;
Misura_Ic = AdcRegs.ADCRESULT5 >> 4;
/* Acquisizione analogica delle correnti di fase (A,B,C) d'ingresso
al motore prese con sonde ad effetto Hall di corrente per la
conversione a digitale, si tolgono i primi 4 bits per condizioni del
ADC */

Ia = Misura_Ia - 0x0800;
Ib = Misura_Ib - 0x0800;
Ic = Misura_Ic - 0x0800;
/* Correnti di fase A, B e C d'ingresso al motore in formato
Q.15(int16),prese con sonde ad effetto Hall di Corrente */

Ib_x_inv_rad_3_ = Ib * _inv_rad_3_;
Ic_x_inv_rad_3_ = Ic * _inv_rad_3_;
/* Prodotto tra le correnti di fase b,c e l'inverso di radice di
3,necessario per il cambio di sistema di coordinate */

I_alpha = Ia;
I_beta = Ib_x_inv_rad_3_ - Ic_x_inv_rad_3_;
/* Correnti nel sistema di riferimento stazionario alpha e beta */

Pr_alpha1 = Sin1 * I_alpha;
Pr_alpha2 = Cos1 * I_alpha;
Pr_beta1 = Sin1 * I_beta;
Pr_beta2 = Cos1 * I_beta;
/* Prodotto necessario per il cambio di sistema di coordinate */
```

```

    Id = Pr_alpha1 - Pr_beta2;
    Iq = Pr_alpha2 + Pr_beta1;
/* Correnti nel sistema di riferimento arbitrario qd */

    Id_mot = Id >> 16;
    Iq_mot = Iq >> 16;
/* Correnti in formato Q.15(int16) nel sistema di riferimento
arbitrario qd */
...

```

Nel caso in cui le correnti di fase sono fisse si programma come in seguito:

```

...
/***** Senza utilizzare l'ADC *****/

    Misura_Ia = 0x00000FFF;
    Misura_Ib = 0x000007FF;
    Misura_Ic = 0x00000000;
/* Correnti di fase (A,B,C) impostata per prova della trasformata */

    Ia = Misura_Ia - 0x0800;
    Ib = Misura_Ib - 0x0800;
    Ic = Misura_Ic - 0x0800;
/* Correnti di fase A, B e C d'ingresso al motore in formato
Q.15(int16), prese con sonde ad effetto Hall di Corrente */

    Ib_x_inv_rad_3_ = Ib * _inv_rad_3_;
    Ic_x_inv_rad_3_ = Ic * _inv_rad_3_;
/* Prodotto tra le correnti di fase b,c e l'inverso di radice di 3,
necessario per il cambio di sistema di coordinate */

    I_alpha = Ia;
    I_beta = Ib_x_inv_rad_3_ - Ic_x_inv_rad_3_;
/* Correnti nel sistema di riferimento stazionario alpha e beta */

    Pr_alpha1 = Sin1 * I_alpha;
    Pr_alpha2 = Cos1 * I_alpha;
    Pr_beta1 = Sin1 * I_beta;
    Pr_beta2 = Cos1 * I_beta;
/* Prodotto necessario per il cambio di sistema di coordinate */

    Id = Pr_alpha1 - Pr_beta2;
    Iq = Pr_alpha2 + Pr_beta1;
/* Correnti nel sistema di riferimento arbitrario qd */

    Id_mot = Id >> 16;
    Iq_mot = Iq >> 16;
/* Correnti in formato Q.15(int16) nel sistema di riferimento
arbitrario qd */
...

```

3.3.7.3. VECTOR TRACKING OBSERVER VTO (Osservatore ad inseguimento vettoriale)

La seguente porzione della routine principale di controllo del motore (interrupt void motorcontrol(void)) è costituita dal VTO. Dal momento che la frequenza di campionamento dell'osservatore è impostata alla quinta parte della frequenza di commutazione della PWM, è necessario disporre di un contatore che permetta

svolgere il VTO una volta ogni 5 esecuzioni della routine motorcontrol(void); per risolvere questo si utilizza un ciclo "if":

```
...
interrupt void motorcontrol(void)
{
    /*##### PULSE-WIDTH MODULATION (PWM)
    #####*/

    /****** Posizione angolare ideale del Rotore del Motore
    *****/

    /****** Struttura della PWM
    *****/

    /*##### ABC → ALPHA,BETA → DQ COORDINATES TRANSFORMATION
    #####*/

    /*##### VECTOR TRACKING OBSERVER VTO
    #####*/

    /****** Posizione angolare ideale del Rotore del Motore
    *****/

    /****** Segnali Halpha e Hbeta delle sonde ad effetto Hall
    *****/

    if (i1==1)
    {
        /****** Algoritmo interno dell'osservatore
        *****/
    }
    i1 = i1 + 1;

    if(i1==6)
        i1 = 1;

    /*##### DIGITAL-ANALOG CONVERTER (DAC)
    #####*/

    return;
}
...
```

Il seguente gruppo di istruzioni corrisponde alla generazione dei Segnali delle sonde ad effetto Hall (si propone questa soluzione poiché ancora si sta in periodo di progettazione, ma al momento della implementazione questi segnali vengono direttamente dalle sonde), dipendente dalla posizione angolare stimata ottenuta con l'osservatore.

```
...
/****** Segnali Halpha e Hbeta delle sonde ad effetto Hall
*****/

if ((theta_hall >= 0xE0000000) & (theta_hall < 0xF0000000))
/*theta_quant_K1 = (pi/4)*/
/*m1 = 0.25*pi in iq29 */
{m1 = 0x1921FB54;
Ha = 0x16A09E66; // Ha = _IQcos(m1)
Hb = 0x16A09E66;} // Hb = _IQsin(m1)
```



```

else if ((theta_hall >= 0xF0000000) & (theta_hall <= 0x00000000))
/*theta_quant_K2 = (3*pi/4)*
/*m2 = 0.75*pi in iq29 */
{m2 = 0x4B65F1FC;
Ha = 0xE95F619A; // Ha = _IQcos(m2)
Hb = 0x16A09E66;} // Hb = _IQsin(m2)

else if ((theta_hall >= 0x00000000) & (theta_hall < 0x10000000))
/*theta_quant_K3 = (-3*pi/4)*
/*m3 = -0.75*pi in iq29 */
{m3 = 0xB49A0E04;
Ha = 0xE95F619A; // Ha = _IQcos(m3)
Hb = 0xE95F619A;} // Hb = _IQsin(m3)

else if ((theta_hall >= 0x10000000) & (theta_hall <= 0x20000000))
/*theta_quant_K4 = (-pi/4)*
/*m4 = -0.25*pi in iq29 */
{m4 = 0xE6DE04AC;
Ha = 0x16A09E66; // Ha = _IQcos(m4)
Hb = 0xE95F619A;} // Hb = _IQsin(m4)

Halpα = _IQtoIQ15(Ha);
Hbeta = _IQtoIQ15(Hb);
/* Halpα(Ha) e Hbeta(Hb) in formato Q.15 per condizioni del DAC*/
...

```

Il segnale di feedforward torque deve essere una stima della coppia di accelerazione netta. Nel caso che sia disponibile, è pertanto opportuno sottrarre la coppia di carico stimata e la coppia elettromagnetica di riferimento. Per la stima della posizione angolare attraverso il VTO non è necessaria la presenza di questa controreazione di coppia, è per questo motivo che all'interno dell'algoritmo è considerato nullo:

```

...
KI_iq29 = _IQ(KI);
I_Ucffo_iq29 = _IQ(I_Ucffo);

torque_feedforward = _IQmpyIQX(KI_iq29,29,I_Ucffo_iq29,29);
/* Torque feedforward, I_Ucffo e KI sono definite nel
DSP281x_Constant.h */
...

```

I guadagni dell'osservatore sono calcolati come descritto nel paragrafo 2.3.3. La larghezza di banda dell'osservatore per il calcolo di ogni guadagno è stata impostata come di seguito:

- $K_1 \rightarrow$ Anello derivativo a 40Hz;
- $K_2 \rightarrow$ Anello proporzionale a 4Hz;
- $K_3 \rightarrow$ Anello integrale a 0,4Hz;

La frequenza di campionamento utilizzata è pari a 15kHz, essendo questa la stessa della PWM. I rispettivi valori dei guadagni definiti nel file DSP281x_Constant.h e la procedura di calcolo degli anelli derivativo-proporzionale-integrale si riportano nel seguito.

DSP281x_Constant.h

```
...
/* Costanti K1, K2 e K3divK2, necessarie per gli anelli derivativo-
proporzionale-integrale, con Ts=1/5kHz */
```

```
#define k_1 0.0066
#define k_2 7.8991e-6
#define k_3divk_2 1.1869e-4
...
```

DSP281x_Osservatore.c

```
...
k_1_iq29 = _IQ(k_1);
/* k_1 è definita nel DSP281x_Constant.h */
K1_obs_error = _IQmpyIQX(k_1_iq29,29,obs_error_1,29);
/* K1_obs_error = k_1_iq29 * obs_error_1 */

k_2_iq29 = _IQ(k_2);
/* k_2 è definita nel DSP281x_Constant.h */
K2_obs_error = _IQmpyIQX(k_2_iq29,29,obs_error_1,29);
/* K2_obs_error = k_2_iq29 * obs_error_1 */

k_3divk_2_iq29 = _IQ(k_3divk_2);
/* k_3divk_2 è definito nel DSP281x_Constant.h */
a = _IQmpyIQX(k_3divk_2_iq29,29,acc_obs_error,29);
/* a = k_3divk_2 * acc_obs_error */
...
```

Dopo aver ottenuto i componenti del vettore Hall rotante ($\vec{H}_{\alpha\beta}$), i guadagni dell'osservatore (K_1 , K_2 e K_3) ed il segnale di feedforward torque (nullo), è ora possibile eseguire il VTO reale. Uno schema a blocchi dell'osservatore con i nomi delle variabili di ingresso ed uscite similari a quelli utilizzati nel codice è mostrato nella figura 3.27, seguita da una discussione dell'algoritmo attraverso l'uso di un pseudo-codice.

La velocità angolare stimata tramite l'osservatore è espresso in per unità rispetto a:

$$(3.6) \quad \omega_{e_max_obs} = \frac{2 \cdot \pi}{T_s}$$

Per implementare (in futuro) l'algoritmo del regolatore di corrente e velocità e poiché il segnale della velocità angolare ottenuta tramite l'osservatore è espresso in per unità rispetto ad una base diversa a ω_{e_max} , è necessario scalare questo valore moltiplicando per $\frac{\omega_{e_max_obs}}{\omega_{e_max}}$. Questo procedimento viene fatto al di fuori del VTO, come si può vedere nella figura 3.17.

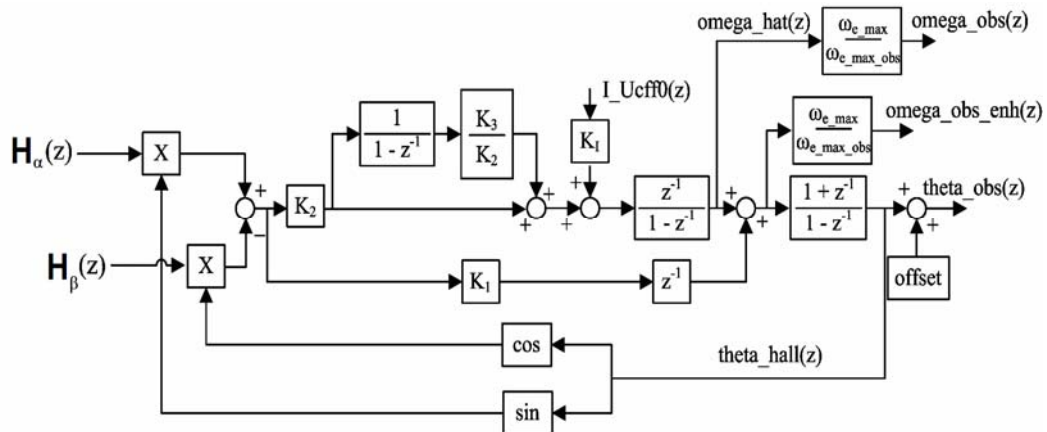


Fig. 3.17. Schema a blocchi del VTO implementato nel software

Il guadagno integrale è espresso come $\frac{K_3}{K_2}$ perché K_3 è molto piccolo. Per quanto

riguarda il formato numerico, per il VTO è stata utilizzata la libreria IQmath, tutte le operazioni sono dichiarate in formato _iq29 (cioè in formato esadecimale 3.29. Quando si moltiplica una variabile in formato _iq29 con un'altra nello stesso formato, il risultato può essere espresso nel formato _iq29 o in qualsiasi altro formato compreso tra il _iq1 e _iq30. Siccome tutte le operazioni si realizzano in formato _iq29 per potere visualizzare le diverse variabili attraverso il DAC e poi sull'oscilloscopio, si passa dal formato _iq29 (GLOBAL_Q) al formato numerico _iq15 (numero composto da 16-bits a virgola fissa con segno, formato richiesto per il DAC) utilizzando la funzione chiamata "IQtoQN".

La posizione angolare di riferimento con cui è confrontata la posizione angolare stimata per l'osservatore viene generata partendo di una velocità costante pari a 400 rad/s, con un Δt pari all'inverso della frequenza dell'onda triangolare u_T di ingresso al comparatore della PWM ($f_T = 15\text{kHz}$).

(3.7)

$$\bar{v}(t) = \frac{d\bar{x}}{dt} \Rightarrow \int_{t_0}^t \bar{v}(t) dt = \int_{x_0(t)}^{\bar{x}(t)} \frac{d\bar{x}}{dt} dt \Rightarrow \left[t_0 = 0 ; \bar{v}(t) = \bar{v}_0(t) = 400 \frac{\text{rad}}{\text{s}} \right] \Rightarrow \bar{x}(t) = \bar{x}_0(t) + \bar{v}_0(t) \cdot t$$

L'integrazione numerica svolta per calcolare la posizione è stata fatta con un accumulatore. E' stato definito il valore di $\bar{v}_0(t) \cdot t$ in per unità nel file DSP281x_Constant.h. La posizione varierà tra 1 e -1, però dopo per essere valutata come argomento del seno e coseno si deve moltiplicare nuovamente per π . La procedura è la seguente:

```

...
/***** Posizione angolare ideale del Rotore del Motore *****/

// Dtheta_prov_10 = _IQ(Dtheta_hex3f_10);
/* Delta theta per una velocità angolare pari a 10rad/s */
// Dtheta_prov_200 = _IQ(Dtheta_hex3f_200);
/* Delta theta per una velocità angolare pari a 200rad/s */

```

```

    Dtheta_prov_400 = _IQ(Dtheta_hex3f_400);
/* Delta theta per una velocità angolare pari a 400rad/s */

// posiz_prov = posiz_prov + Dtheta_prov_10;
/* Posizione angolare di riferimento, con Velocità Angolare 10rad/s
*/
// posiz_prov = posiz_prov + Dtheta_prov_200;
/* Posizione angolare di riferimento, con Velocità Angolare 200rad/s
*/
    posiz_prov = posiz_prov + Dtheta_prov_400;
/* Posizione angolare di riferimento, con Velocità Angolare 400rad/s
*/

    if (posiz_prov >= uno_iq29)
        posiz_prov = menouno_iq29;
/* Ciclo creato per impostare la posizione angolare stimata (in per
unità) tra -1 e +1 */

    posiz_prov_vis = _IQtoIQ15(posiz_prov);
/* posiz_prov in formato Q.15 per condizioni del DAC*/
...

```

Il seguente pseudo-codice è l'algoritmo equivalente dello schema a blocchi della figura 3.27; i nomi delle variabili e costanti che vengono utilizzati sono simili a quelli usati nel programma riportato nell'appendice, ma il codice è semplificato per mostrare lo schema delle operazioni eseguite.

```

...
    cos_theta_hall = cos(theta_hall);
    sin_theta_hall = sin(theta_hall);
    error_alpha = halpha * sin_theta_hall;
    error_beta = hbeta * cos_theta_hall;
    obs_error = error_alpha - error_beta;

    k2_obs_error = k2 * obs_error;
    acc_obs_error = acc_obs_error + k2_obs_error;
    int_plus_prop = k2_obs_error + k3divk2 * acc_obs_error;
    torque_feedforward = I_Ucfft * KI;

    omega_dot_hat = int_plus_prop + torque_feedforward;
    omega_hat = omega_hat + omega_dot_hat;
    theta_hall = theta_hall + omega_hat_enh;
    omega_hat_enh = omega_hat + k1_obs_error;
    theta_hall = theta_hall + omega_hat_enh;

    k1_obs_error = k1 * obs_error;
...

```

3.3.7.4. Digital-Analog Converter DAC (Convertitore Digitale-Analogico)

Alla fine della routine generale motorcontrol(void) è stata inserita questa routine che rende possibile identificare le variabili da convertire con il Convertitore Digitale-Analogico (AD7568). Un massimo di 8 variabili si possono inviare al DAC per poter essere visualizzati da un oscilloscopio. Questo convertitore insieme al DSP eZdsp™ TMS320F2812 si trova sopra una scheda fatta apposta per coordinare tutto il sistema di controllo del motore, su questa scheda si trovano anche i pins che si

utilizzano per inviare i segnali PWM all'inverter di potenza 3-fase. Questa routine è molto utile durante lo sviluppo del software e nella fase di debug; solo ammette variabili in formato Q.15, è per questo motivo che si utilizza la funzione della libreria IQMATH che converte le variabili dichiarate come GLOBAL_Q al formato Q.15. L'algoritmo utilizzato per l'invio dei dati al DAC è mostrato da seguito:

```
...
/*##### DIGITAL-ANALOG CONVERTER (DAC)
#####*/

DAC_buffer[0] = 0x4000; // data to be displayed on DAC channel 1
DAC_buffer[1] = 0xFFFF; // data to be displayed on DAC channel 2
DAC_buffer[2] = 0xC000; // data to be displayed on DAC channel 3
DAC_buffer[3] = 0x0000; // data to be displayed on DAC channel 4
DAC_buffer[4] = 0x7FFF; // data to be displayed on DAC channel 5
DAC_buffer[5] = 0xE000; // data to be displayed on DAC channel 6
DAC_buffer[6] = 0xA000; // data to be displayed on DAC channel 7
DAC_buffer[7] = 0x6000; // data to be displayed on DAC channel 8

GpioDataRegs.GPFSET.bit.GPIOF6 = 1;    // set CANTX pin
AD7568_write(DAC_buffer);
GpioDataRegs.GPFCLEAR.bit.GPIOF6 = 1;  // clear CANTX pin
...
```

Nell'appendice si trova un algoritmo simile a questo con la differenza che i dati ad essere convertiti sono le variabili del VTO.

3.4. Conclusioni

La struttura del software per il controllo di posizione del motore ed il codice che viene utilizzato nel DSP è stato presentato in questo capitolo. All'interno di questo software si trovano varie applicazioni dove la più importante è quella riguardante il VTO. Il codice nella sua maggioranza è stato scritto in linguaggio C/C++ e solo qualche istruzione è stata scritta in linguaggio assembler; il codice è stato spiegato attraverso l'uso di parti reali del software, diagrammi a blocchi e pseudo-codice. Importanti caratteristiche di implementazione includono l'uso di aritmetica a 32-bits per migliorare la precisione nei calcoli nel VTO; con aiuto in alcuni casi delle librerie Qmath (PWM) ed IQMATH (VTO). Si genera una funzione che proporziona la posizione angolare di riferimento con cui si confronta la posizione angolare stimata ottenuta dall'osservatore. Per ultimo, le variabili del VTO sono visualizzate nell'oscilloscopio attraverso l'utilizzo del Convertitore Digitale-Analogico.

Capitolo 4

Simulazioni e prove sperimentali sull'osservatore

4.1. Introduzione

Questo capitolo presenta i risultati sperimentali per convalidare il metodo proposto nel Capitolo 2 per la stima della posizione e velocità angolare. Inoltre si presentano i risultati dell'implementazione dell'osservatore ad inseguimento vettoriale (VTO) nel DSP eZdsp™ TMS320F2812 descritto nel Capitolo 3. Il paragrafo 4.2 descrive l'implementazione dell'osservatore attraverso Matlab, mentre i risultati delle simulazione si trovano nel paragrafo 4.3. Nel paragrafo 4.4 si descrive l'impostazione dell'osservatore sul DSP eZdsp™ TMS320F2812, mentre nel paragrafo 4.5 si mostrano i risultati sperimentali delle prove all'osservatore attraverso il DSP eZdsp™ TMS320F2812. Per finire questo capitolo, nel paragrafo 4.6 si fa un confronto tra i risultati delle simulazioni e quelli delle prove sperimentali.

4.2. Impostazione dell'osservatore attraverso Matlab

Con l'intenzione di studiare il comportamento ideale dell'osservatore ad inseguimento vettoriale (VTO), è stato realizzato su Simulink (Matlab) il diagramma a blocchi della figura 4.1, il quale è una rappresentazione del VTO.

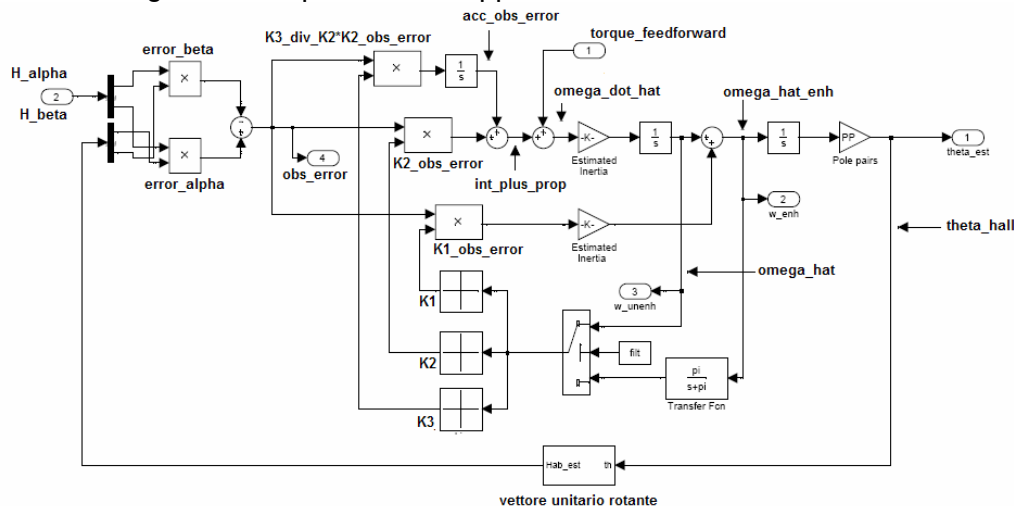


Fig. 4.1. Diagramma a blocchi del VTO su Simulink (Matlab)

Il VTO è impostato con le seguenti caratteristiche:

- Larghezza di banda pari a 40, 4 e 0,4 Hz;
- Inerzia stimata $\hat{J}$ pari a 0,018 Kg.m²;
- Numero di coppie polari "p" pari a 4;
- Velocità Angolare Ideale di 10, 200 e 400 rad/s, con cui si riproducono le posizione angolare ideale per ogni valore di velocità.
- Senza utilizzare la Gain Scheduling e disaccoppiamento delle armoniche.

4.3. Risultati delle simulazioni attraverso Matlab

Utilizzando il diagramma a blocchi mostrato nel paragrafo precedente, sono state eseguite 3 tipi di simulazione (Velocità Angolare 10, 200 e 400 rad/s) ed i risultati ottenuti sono di seguito riportati.

4.3.1. Posizione Angolare Stimata (Velocità Angolare 10 rad/s)

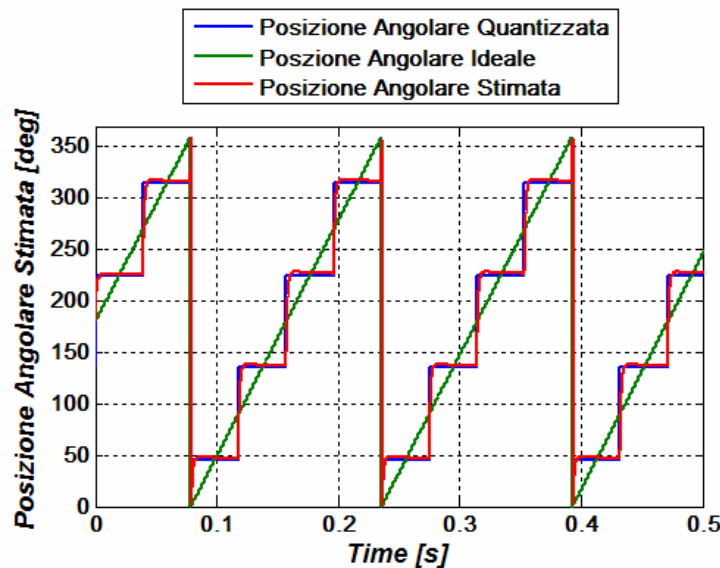


Fig. 4.2. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

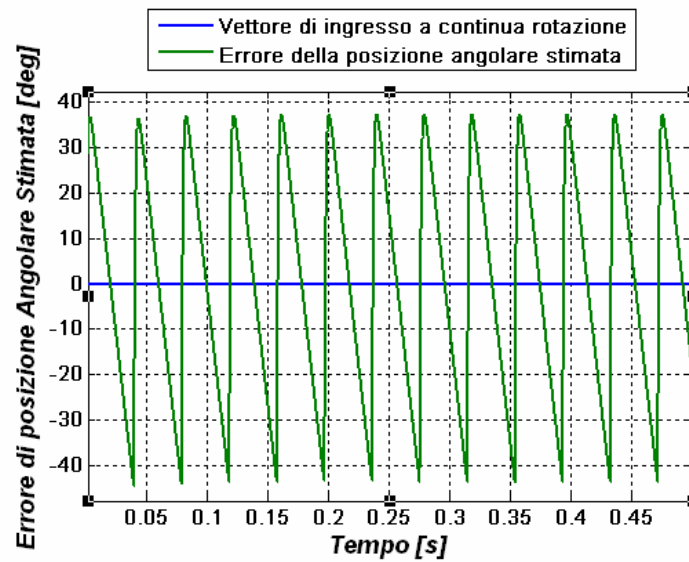


Fig. 4.3. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

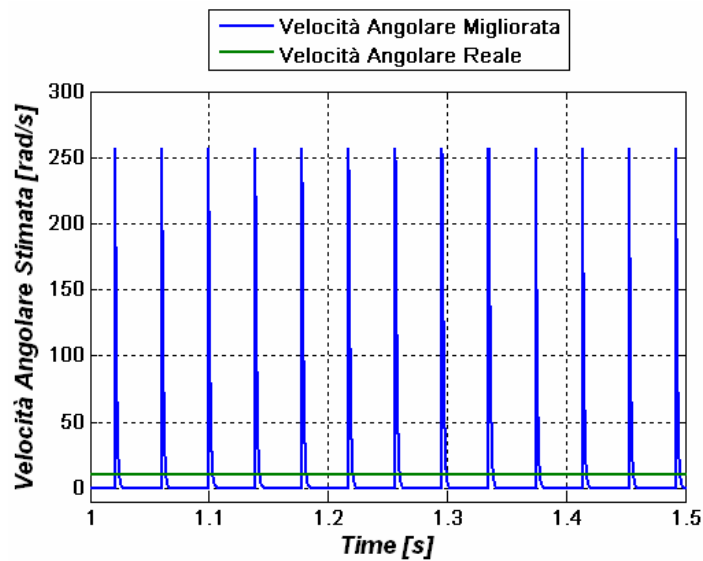


Fig. 4.4. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

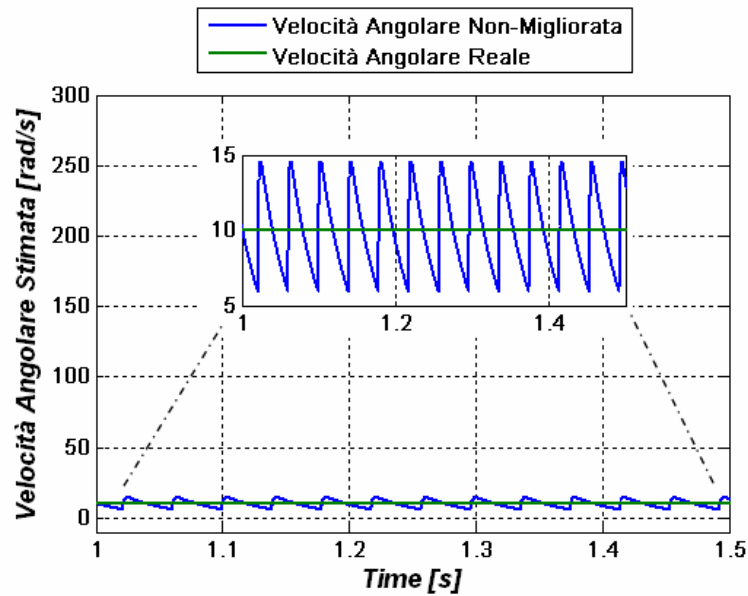


Fig. 4.5. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

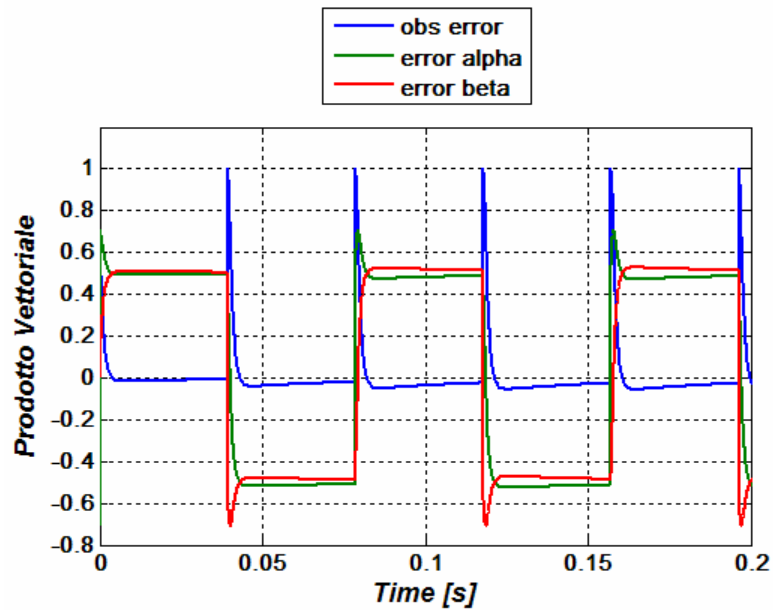


Fig. 4.6. Simulazione del obs_error, error_alpha e error_beta ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

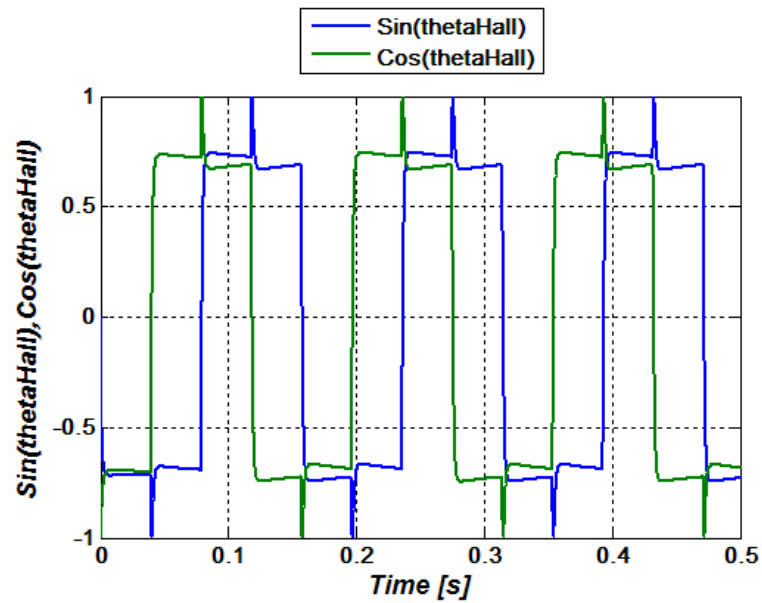


Fig. 4.7. Simulazione del $\text{Sin}(\theta_{\text{HALL}})$ e $\text{Cos}(\theta_{\text{HALL}})$ ($\omega_{\text{el}}=10 \text{ rad/s}$; $\text{BW} = 40,4,0.4 \text{ Hz}$)

4.3.2. Posizione Angolare Stimata (Velocità Angolare 200 rad/s)

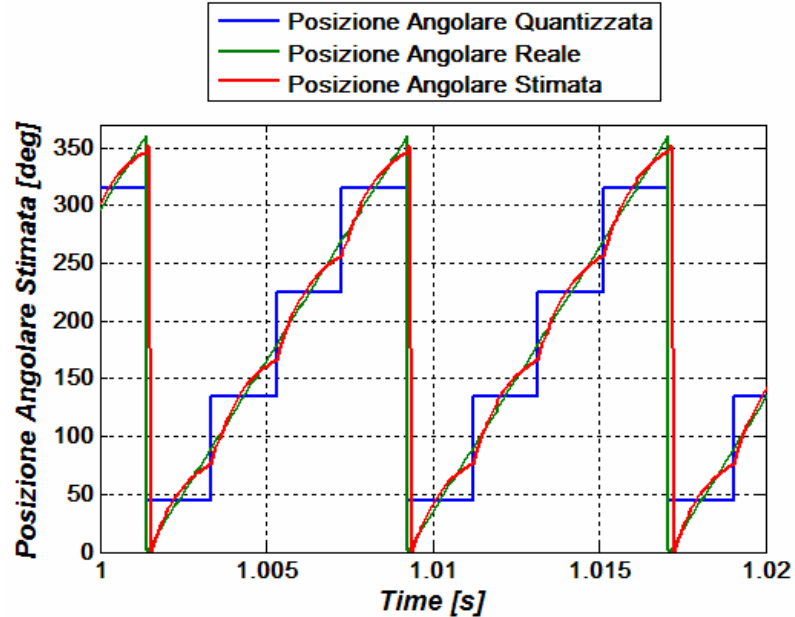


Fig. 4.8. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{\text{el}}=200 \text{ rad/s}$; $\text{BW} = 40,4,0.4 \text{ Hz}$)

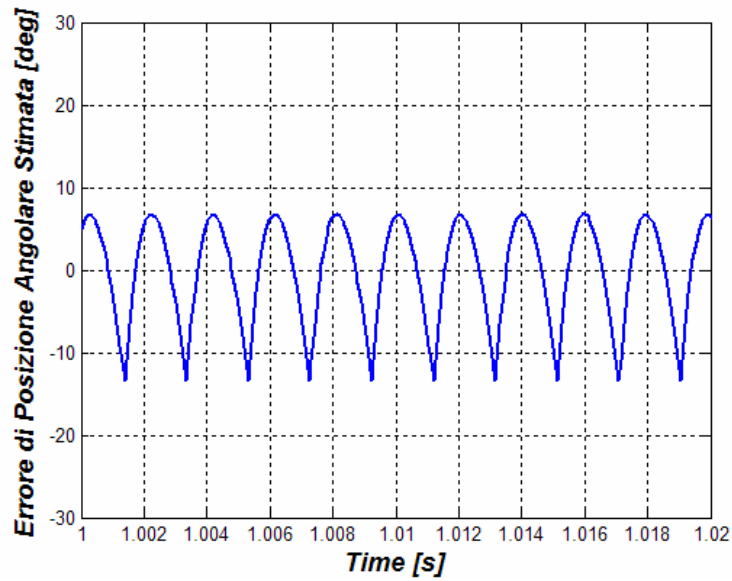


Fig. 4.9. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

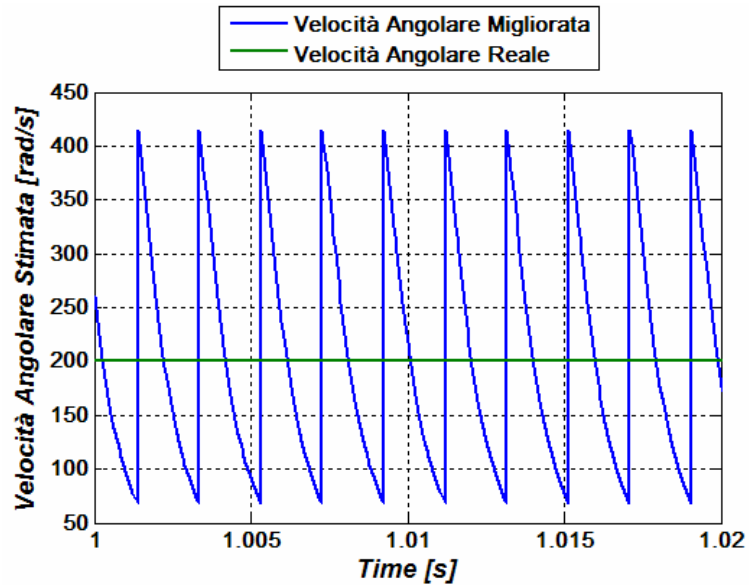


Fig. 4.10. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

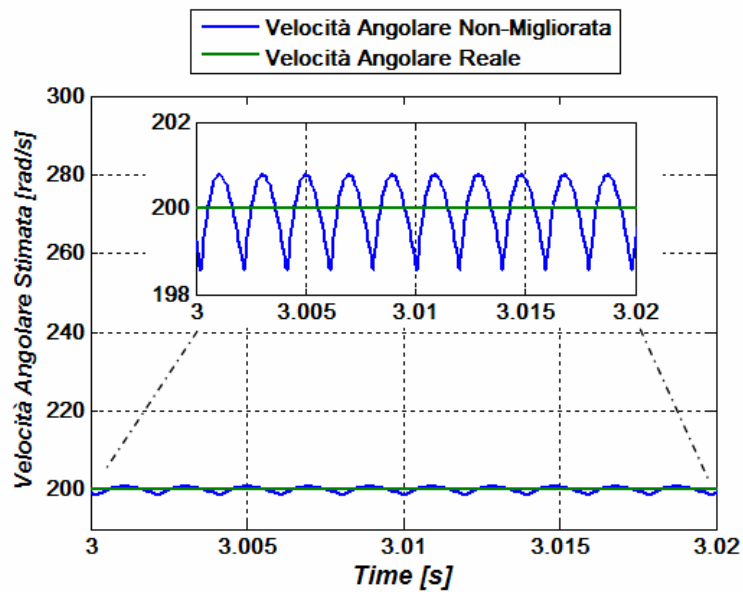


Fig. 4.11. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

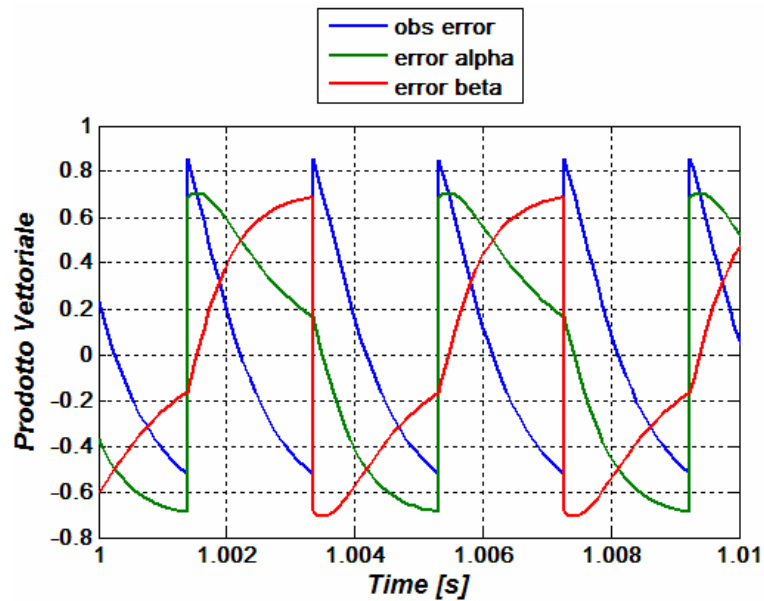


Fig. 4.12. Simulazione del obs_error, error_alpha e error_beta ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

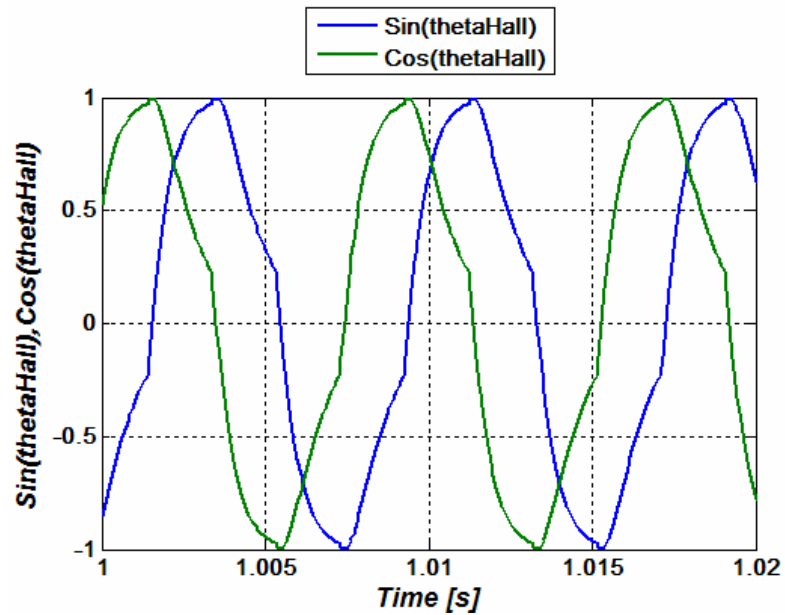


Fig. 4.13. Simulazione del $\text{Sin}(\theta_{\text{HALL}})$ e $\text{Cos}(\theta_{\text{HALL}})$ ($\omega_{el}=200 \text{ rad/s}$; $BW = 40,4,0.4 \text{ Hz}$)

4.3.3. Posizione Angolare Stimata (Velocità Angolare 400 rad/s)

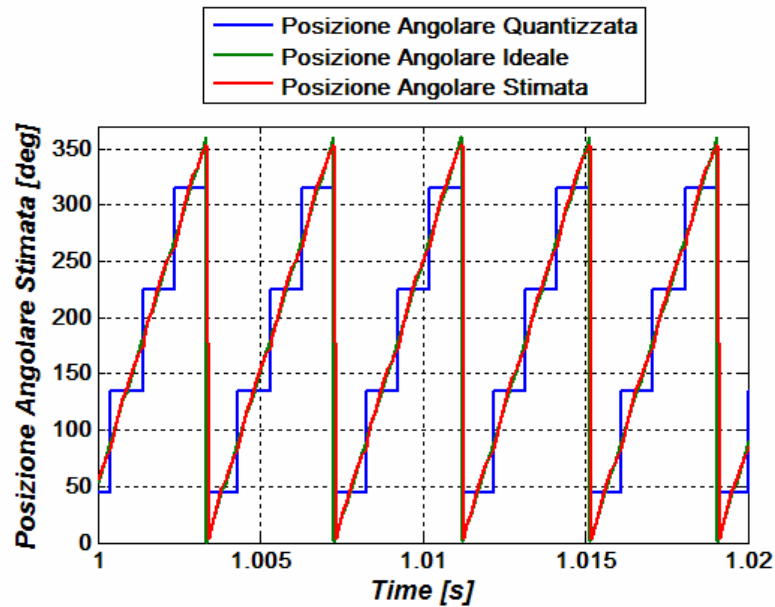


Fig. 4.14. Simulazione in regime stazionario della posizione quantizzata, ideale e stimata ($\omega_{el}=400 \text{ rad/s}$; $BW = 40,4,0.4 \text{ Hz}$)

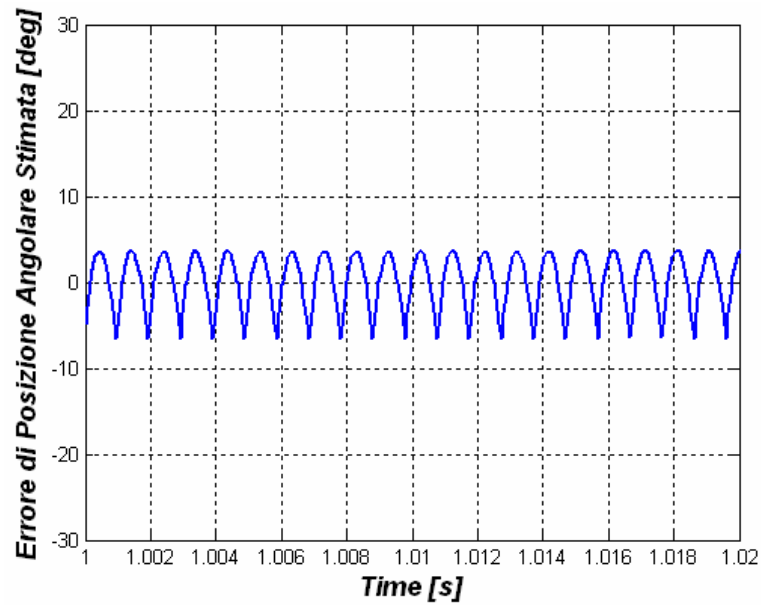


Fig. 4.15. Simulazione in regime stazionario dell'errore di posizione angolare stimata ($\omega_{el}=400$ rad/s ; $BW = 40,4,0.4$ Hz)

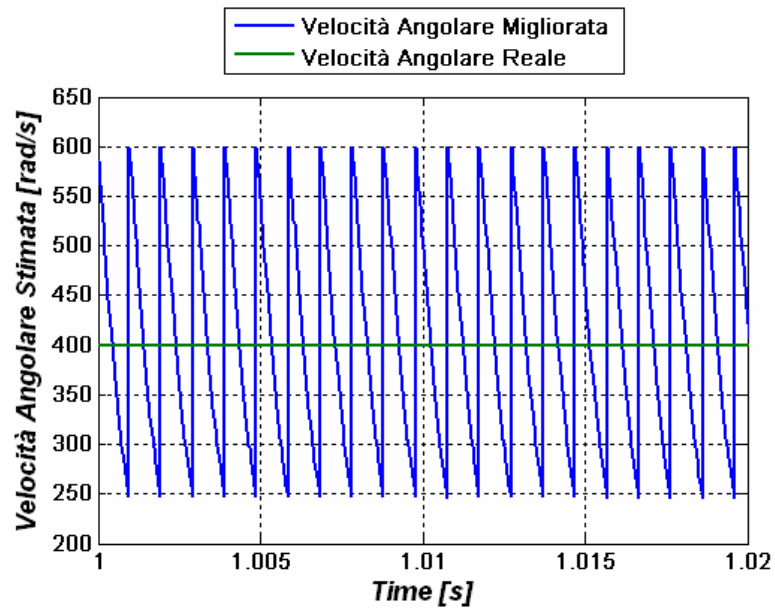


Fig. 4.16. Simulazione della velocità angolare reale e la stima della velocità angolare migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; $BW = 40,4,0.4$ Hz)

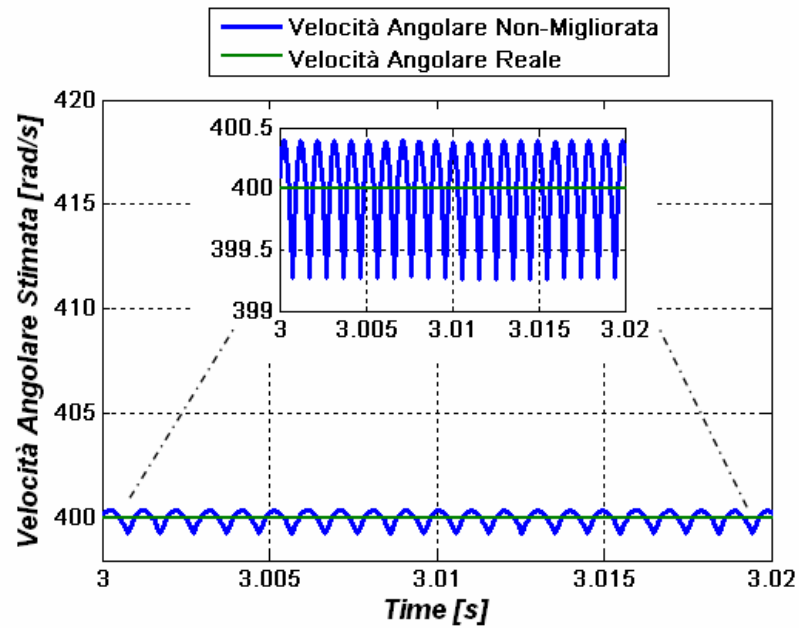


Fig. 4.17. Simulazione della velocità angolare reale e la stima della velocità angolare non-migliorata a regime stazionario ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

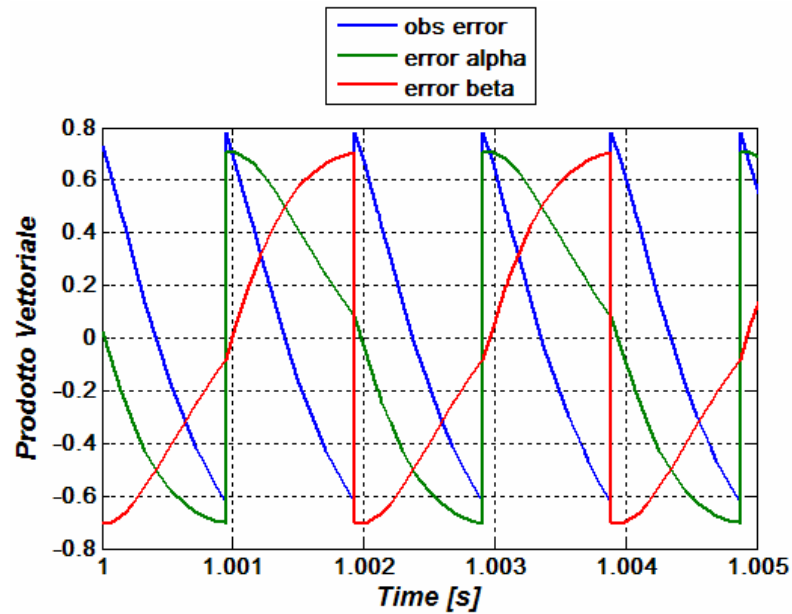


Fig. 4.18. Simulazione del obs_error, error_alpha e error_beta ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

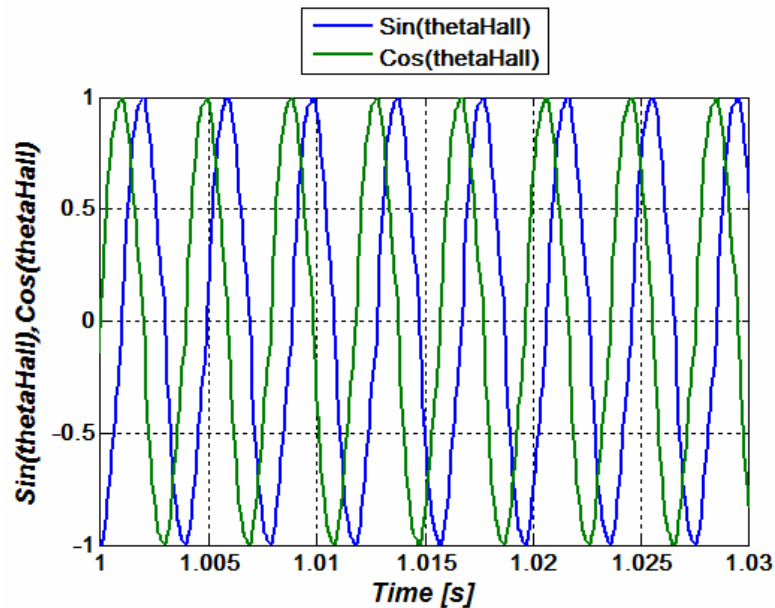


Fig. 4.19. Simulazione del $\text{Sin}(\theta_{\text{HALL}})$ e $\text{Cos}(\theta_{\text{HALL}})$ ($\omega_{\text{el}}=400 \text{ rad/s}$; $\text{BW} = 40,4,0.4 \text{ Hz}$)

4.4. Impostazione dell'osservatore sul DSP eZdsp TMS320F2812

La struttura sperimentale utilizzata per implementare il VTO ed ottenere la posizione e velocità angolare stimata attraverso l'hardware e software del DSP si mostra da seguito.

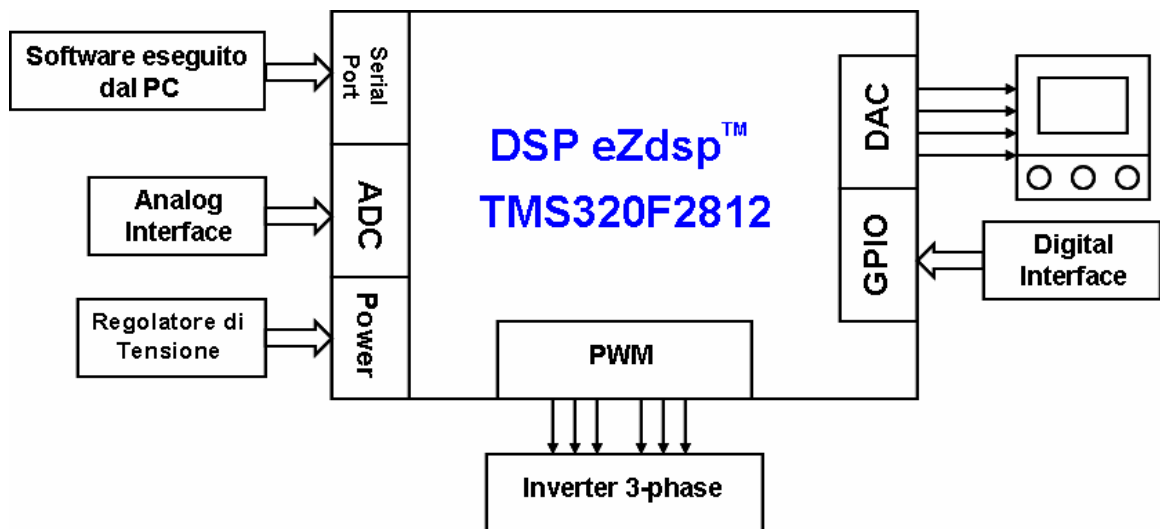


Fig. 4.20. Diagramma a blocchi della struttura sperimentale per implementare il VTO

L'hardware incaricato di fare il sistema di controllo è la scheda DSP eZdsp TMS320F2812, la quale è stata descritta nel paragrafo 3.2. Il software di controllo (Code Composer Studio– Platinum Edition) è eseguito dal PC e inviato alla scheda mediante la porta seriale. Per l'acquisizione dei segnali analogici si utilizza il

convertitore analogico-digitale (ADC), l'alimentazione della scheda in questo caso si fa mediante un regolatore di tensione. Le uscite PWM per comandare l'inverter 3-phase sono disponibile nel P8 (I/O interface); nel caso di acquisizioni digitali si utilizza la General-Purpose Inputs/Outputs (GPIO). Per visualizzare le variabili del software disponibile nel DAC (Digital-Analog Converter) si utilizza un oscilloscopio digitale.



Fig. 4.21. Struttura sperimentale per implementare il VTO

4.5. Risultati sperimentali delle prove all'osservatore attraverso il DSP eZdsp TMS320F2812

In questo paragrafo si mostrano i risultati sperimentali del VTO ottenuti attraverso il DSP. Nei paragrafi 4.5.1, 4.5.2 e 4.5.3 si mostrano le prestazioni del VTO nella stima della posizione angolare e velocità angolare, quando si utilizzano 3 diverse velocità angolari di riferimento; in questo caso non si utilizzano i metodi per migliorare l'inseguimento vettoriale proposti nei paragrafi 2.4.2 e 2.4.1 (Gain Scheduling e disaccoppiamento delle armoniche). Nel paragrafo 4.5.4 si mostra il segnale PWM (ad una frequenza di 15kHz) per comandare l'inverter generato attraverso la routine descritta nel paragrafo 3.3.7.1; mentre nel paragrafo 4.5.5 si mostrano le correnti nel sistema di riferimento arbitrario (qd0) ottenute da 3 valori fissi di corrente di fase, la routine utilizzata è stata descritta nel paragrafo 3.3.7.2.

4.5.1. Posizione Angolare Stimata (Velocità Angolare 10 rad/s)

Le seguenti figure rappresentano le prestazioni del VTO per l'inseguimento della posizione angolare di riferimento, quando la velocità angolare di riferimento è impostata a 10 rad/s.

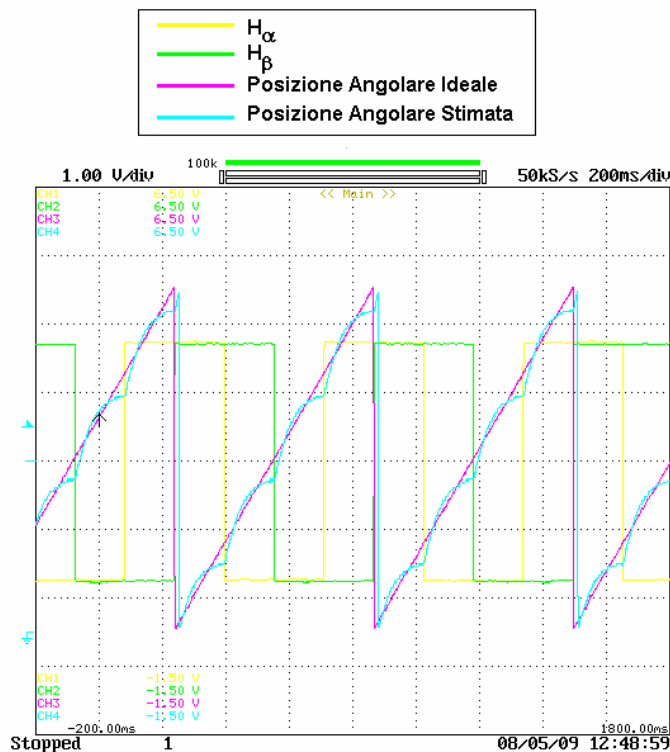


Fig. 4.22. Componenti del vettore Hall rotante (H_α, H_β), la posizione angolare ideale e stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

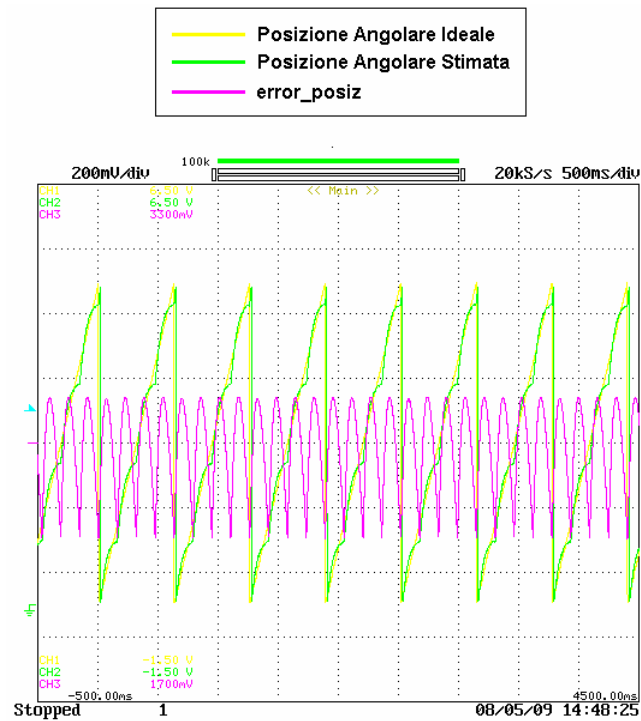


Fig. 4.23. Errore di posizione angolare stimata, la posizione angolare ideale e stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

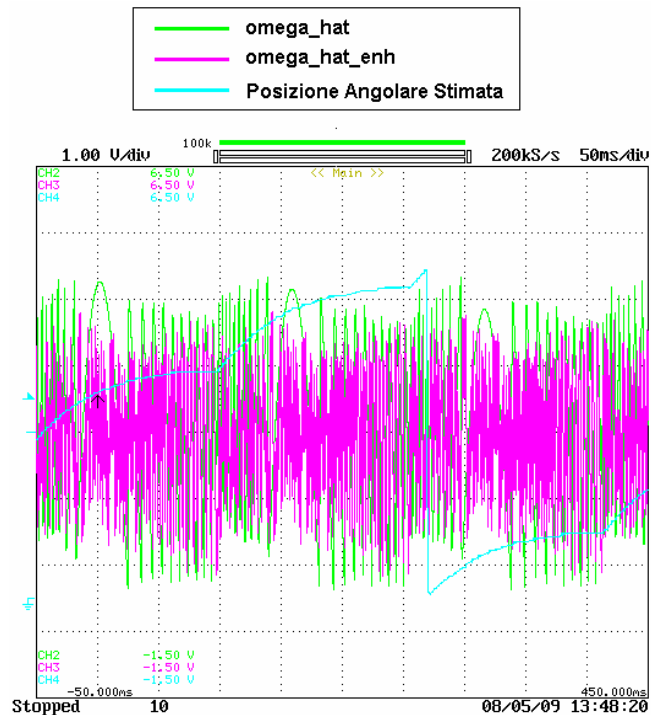


Fig. 4.24. Velocità angolare migliorata, non-migliorata e la posizione angolare stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

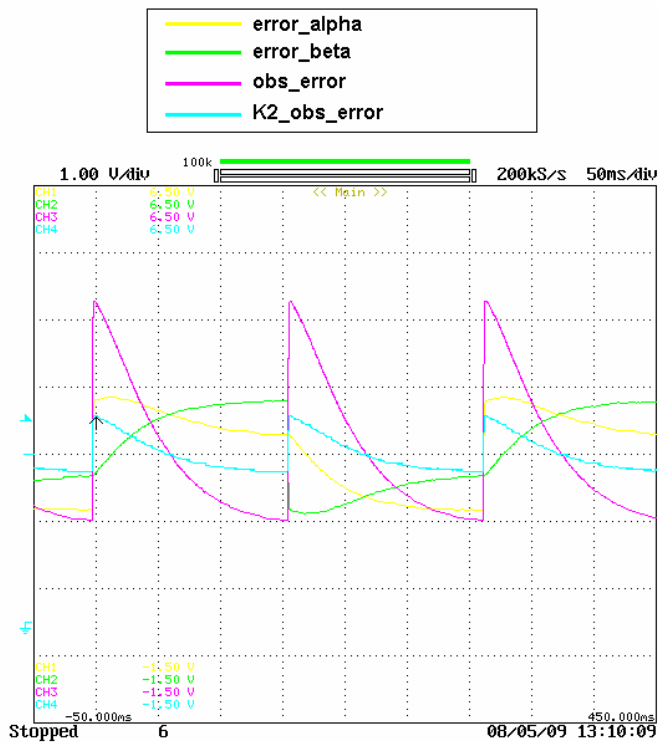


Fig. 4.25. Prodotto vettoriale, errore prodotto vettoriale (obs_error) ed costante del anello proporzionale ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

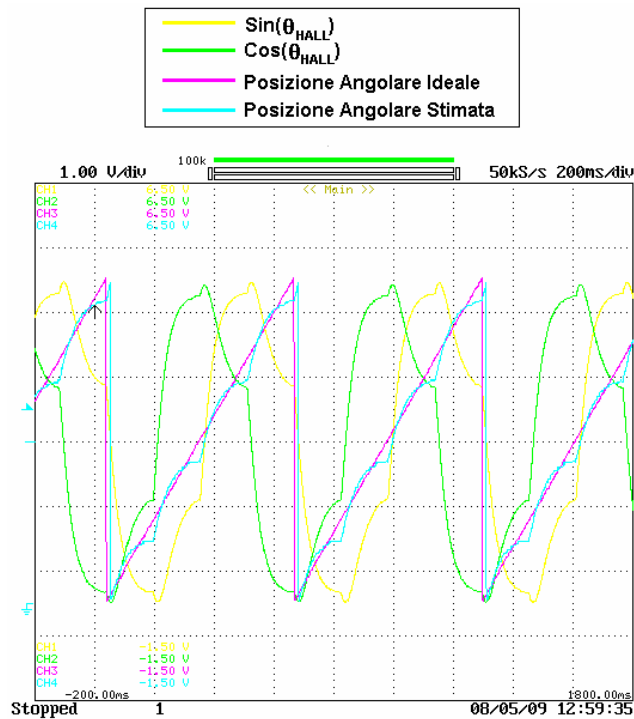


Fig. 4.26. Componenti del vettore unitario rotanti, la posizione angolare ideale e stimata ($\omega_{el}=10$ rad/s ; BW = 40,4,0.4 Hz)

4.5.2. Posizione Angolare Stimata (Velocità Angolare 200 rad/s)

Le seguenti figure rappresentano le prestazioni del VTO per l'inseguimento della posizione angolare di riferimento, quando la velocità angolare di riferimento è impostata a 200 rad/s.

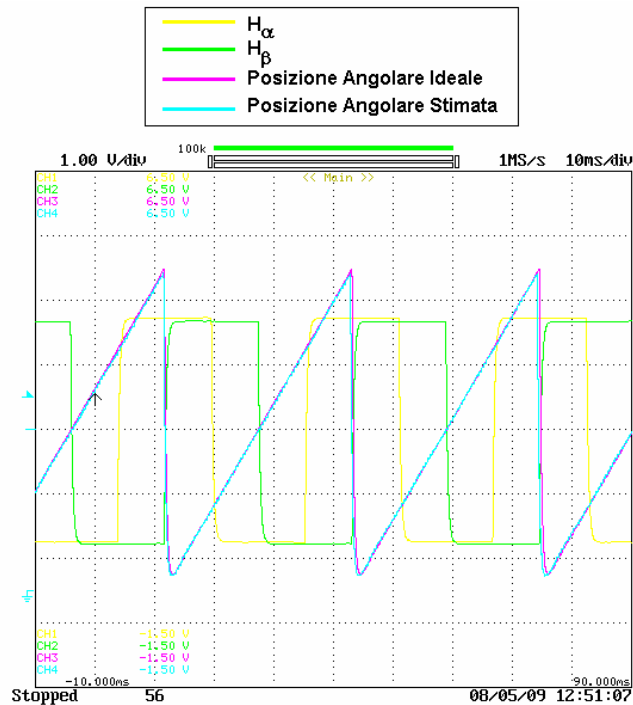


Fig. 4.27. Componenti del vettore Hall rotante (H_α, H_β), la posizione angolare ideale e stimata ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

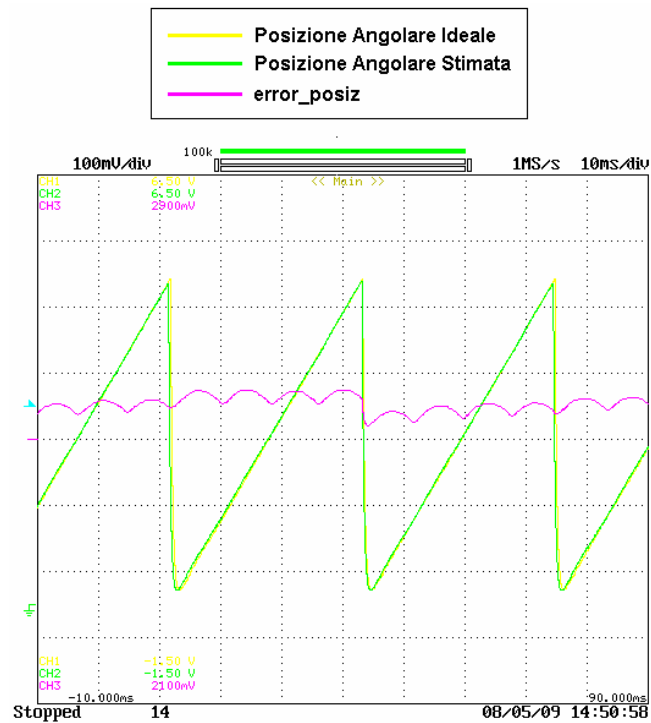


Fig. 4.28. Errore di posizione angolare stimata, la posizione angolare ideale e stimata ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

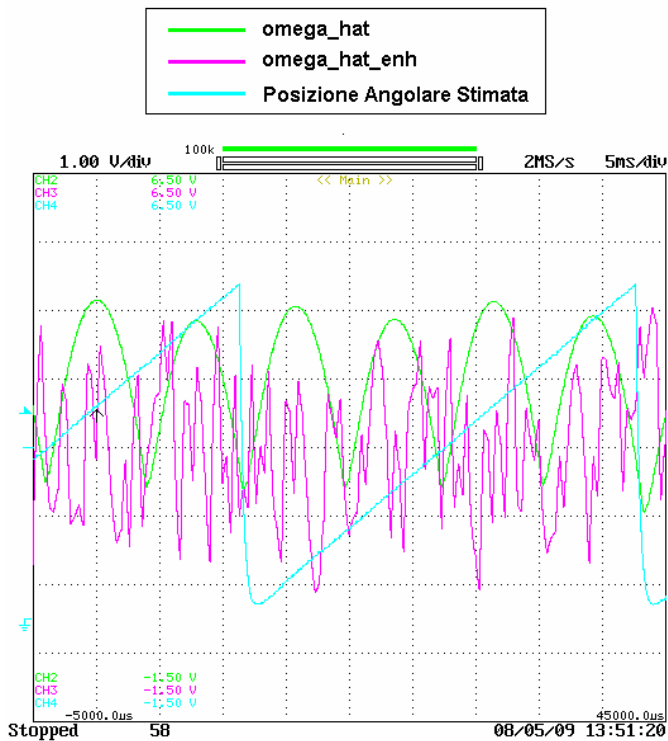


Fig. 4.29. Velocità angolare migliorata, non-migliorata e la posizione angolare stimata ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

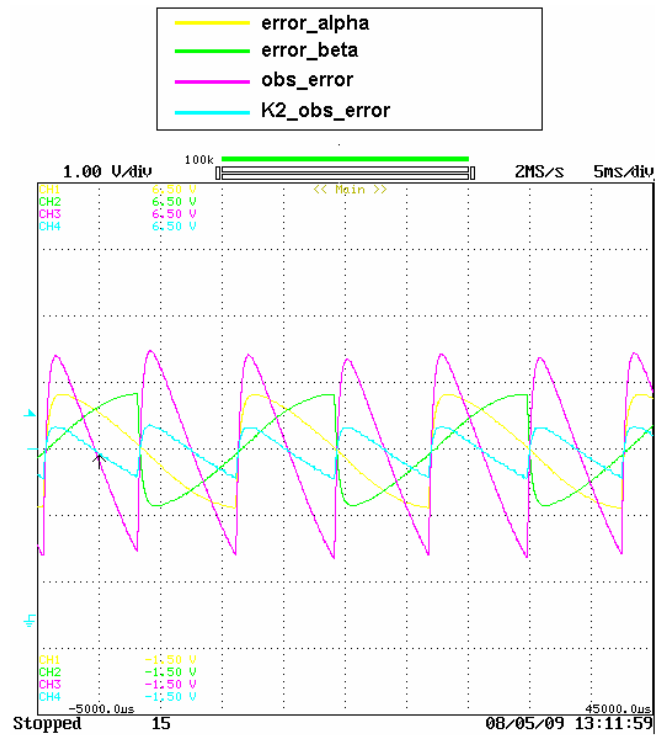


Fig. 4.30. Prodotto vettoriale, errore prodotto vettoriale (obs_error) ed costante del anello proporzionale ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

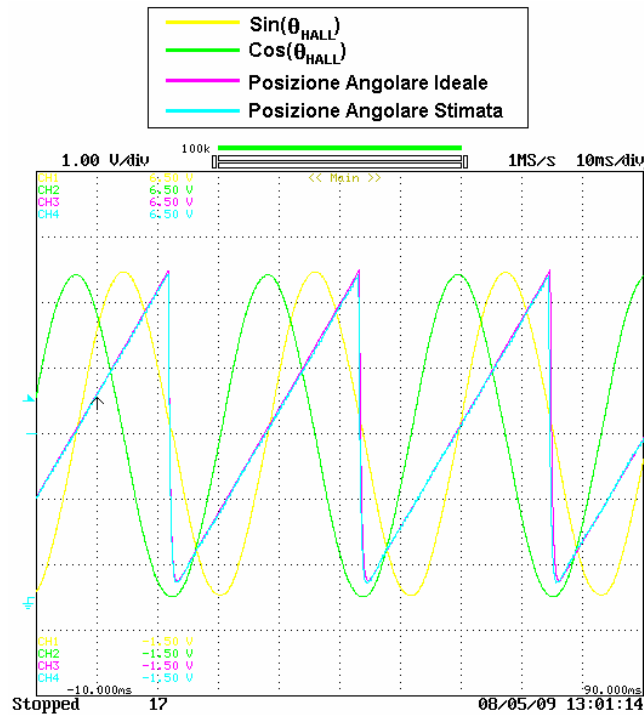


Fig. 4.31. Componenti del vettore unitario rotanti, la posizione angolare ideale e stimata ($\omega_{el}=200$ rad/s ; BW = 40,4,0.4 Hz)

4.5.3. Posizione Angolare Stimata (Velocità Angolare 400 rad/s)

Le seguenti figure rappresentano le prestazioni del VTO per l'inseguimento della posizione angolare di riferimento, quando la velocità angolare di riferimento è impostata a 400 rad/s.

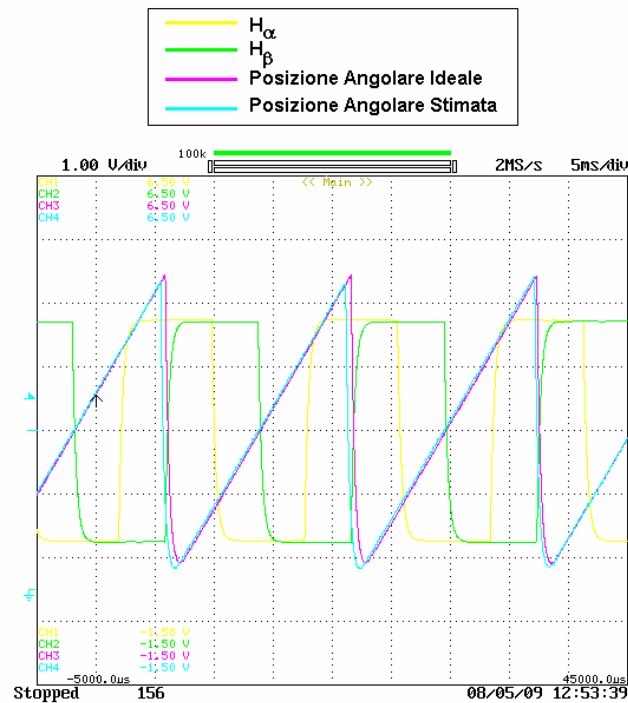


Fig. 4.32. Componenti del vettore Hall rotante (H_α, H_β), la posizione angolare ideale e stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

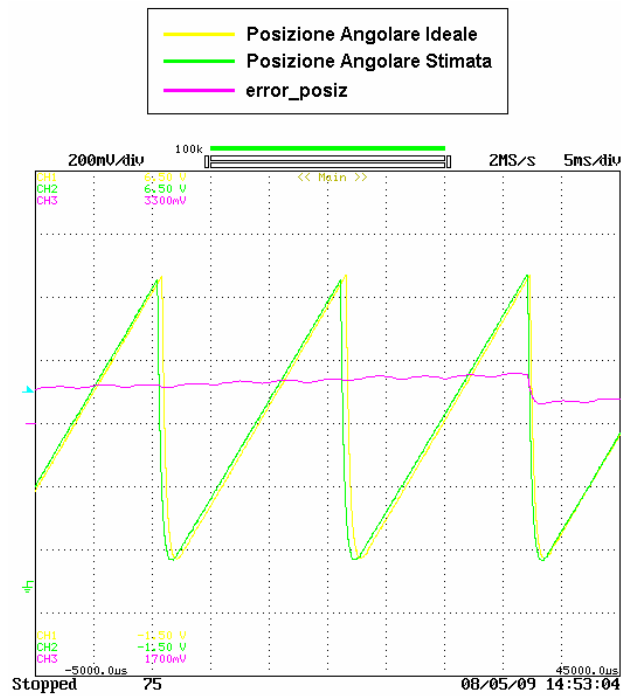


Fig. 4.33. Errore di posizione angolare stimata, la posizione angolare ideale e stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

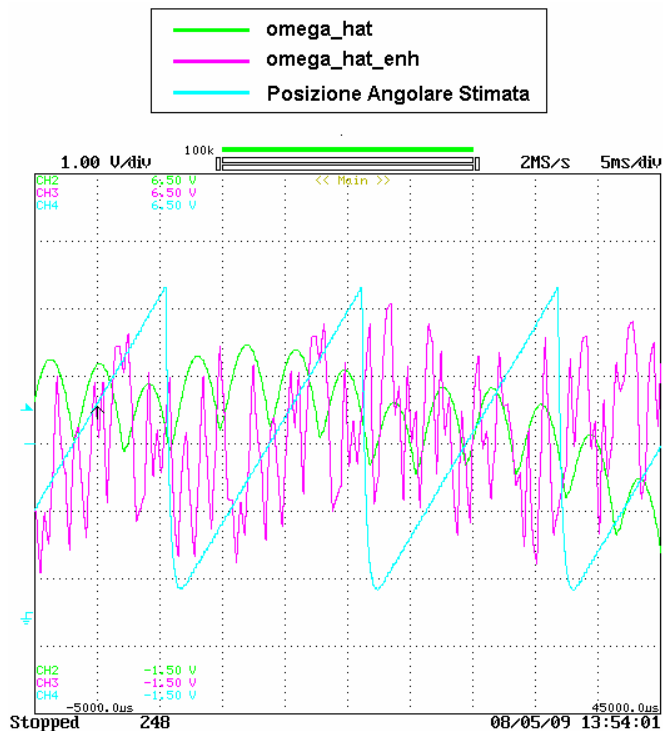


Fig. 4.34. Velocità angolare migliorata, non-migliorata e la posizione angolare stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

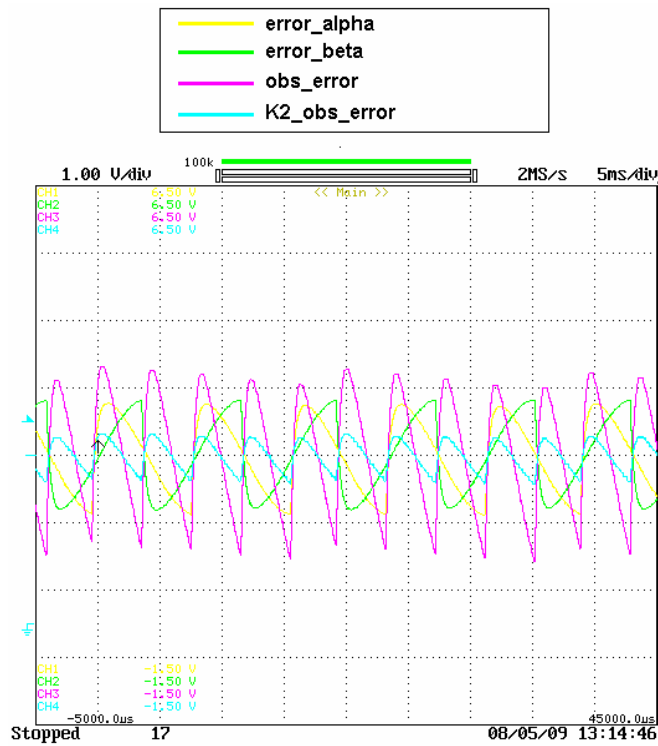


Fig. 4.35. Prodotto vettoriale, errore prodotto vettoriale (obs_error) ed costante del anello proporzionale ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

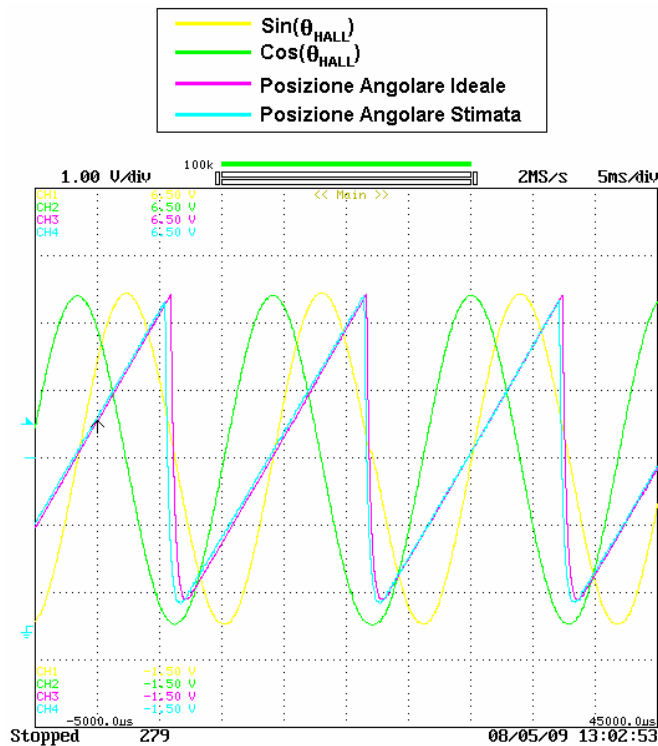


Fig. 4.36. Componenti del vettore unitario rotanti, la posizione angolare ideale e stimata ($\omega_{el}=400$ rad/s ; BW = 40,4,0.4 Hz)

4.5.4. Pulse-Width Modulation PWM

Il segnale PWM è stato generato ad una frequenza di 15kHz e senza l'utilizzo dell'ADC, cioè l'ampiezza della modulante è fissa. Il risultato si mostra da seguito:

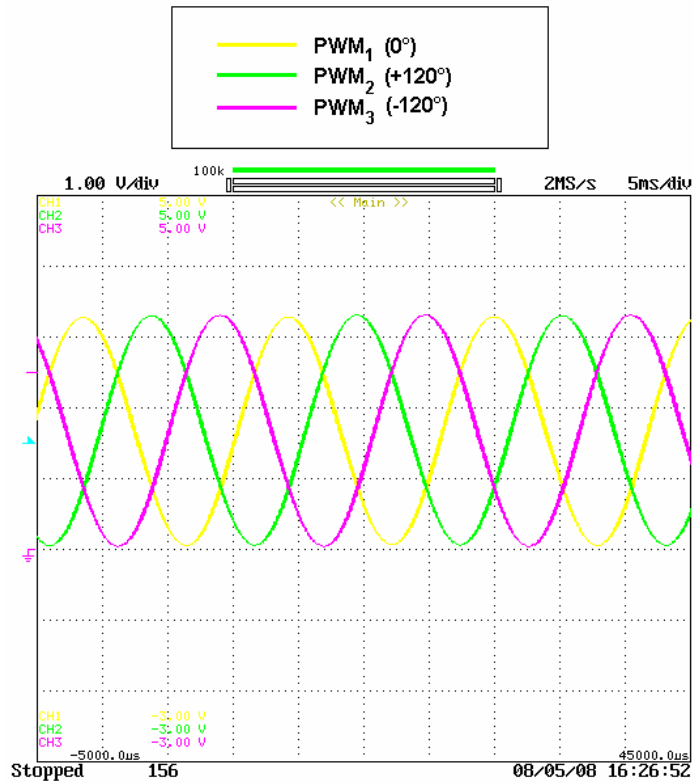


Fig. 4.37. Segnale PWM per comandare l'inverter trifase filtrato a 500Hz

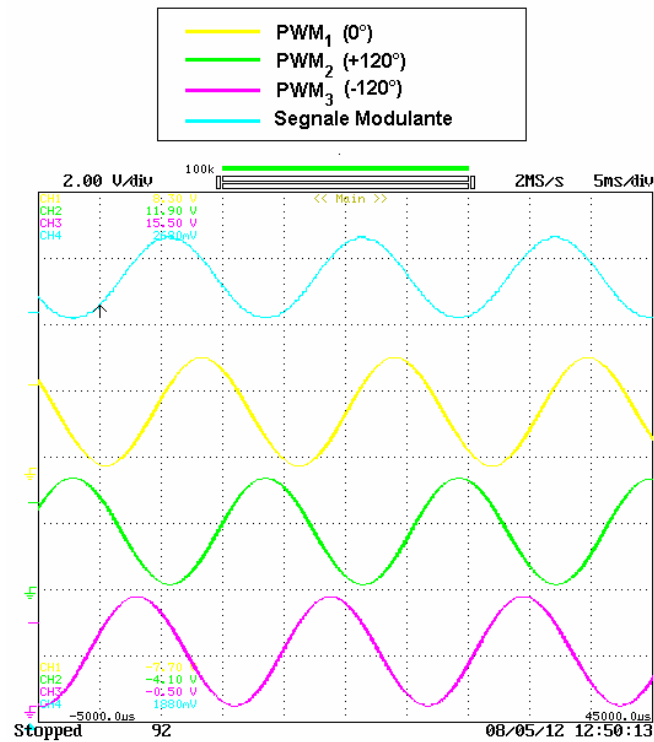


Fig. 4.38. Segnale Modulante e PWM per comandare l'inverter trifase filtrato a 500Hz

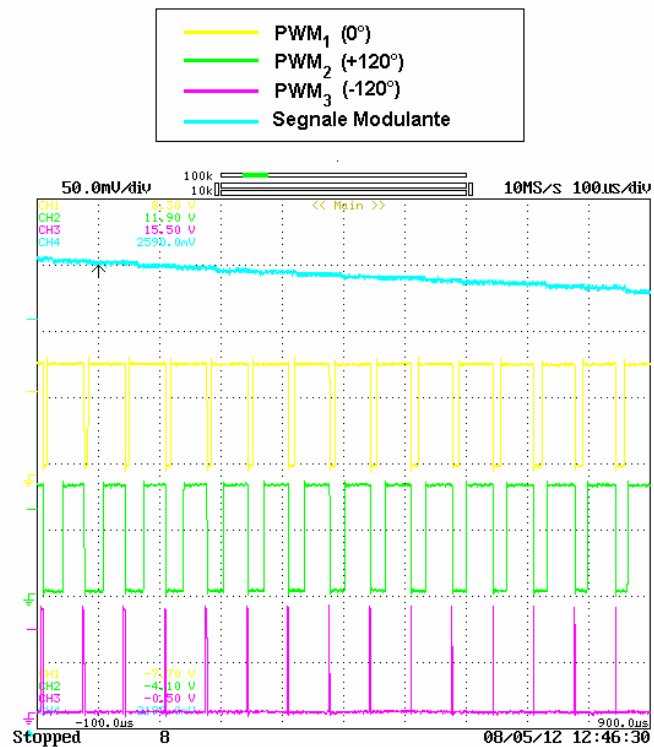
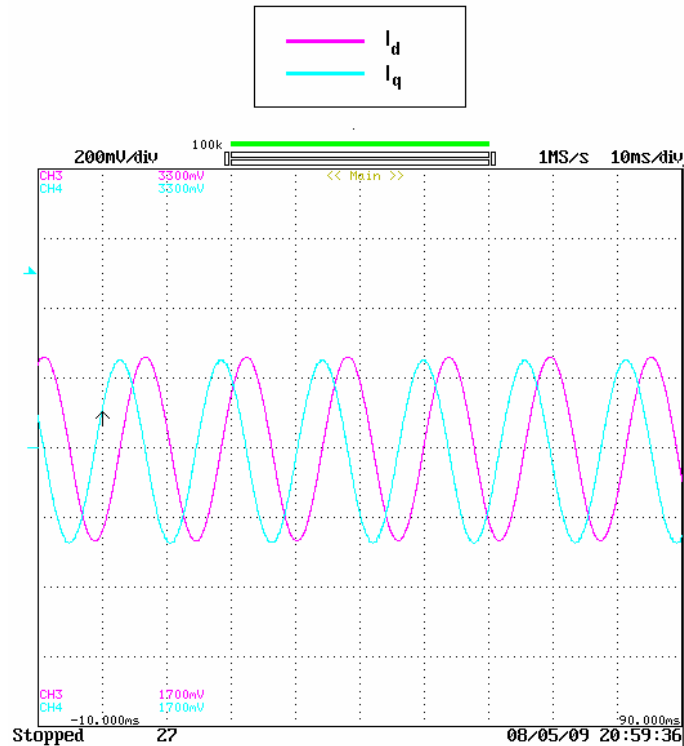


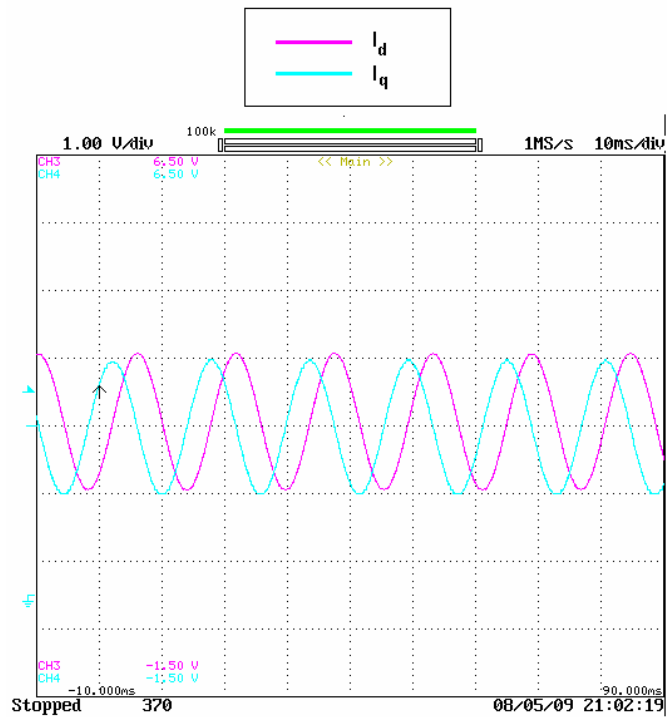
Fig. 4.39. Segnale Modulante e PWM per comandare l'inverter trifase non FILTRATO

4.5.5. Corrente nel sistema di riferimento arbitrario (qd0)

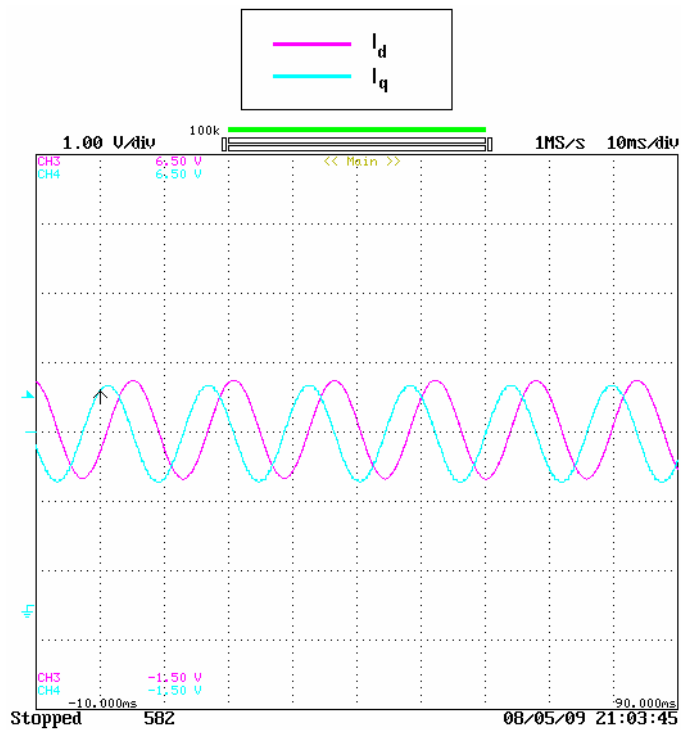
Questa prova è stata effettuata senza l'utilizzo dell'ADC e quindi per vedere l'applicazione si hanno impostato le correnti di fase (I_a , I_b e I_c) a valori costanti, ci sono 3 gruppi di valori ed i risultati si mostrano da seguito.



4.38. Corrente nel sistema di riferimento arbitrario (qd0)
 ($I_a=0x0000FFF$, $I_b=0x000007FF$ e $I_c=0x00000000$)



4.39. Corrente nel sistema di riferimento arbitrario (qd0)
 $(I_a=0x00000B00, I_b=0x000007FF \text{ e } I_c=0x00000500)$



4.40. Corrente nel sistema di riferimento arbitrario (qd0)
 $(I_a=0x00000AAA, I_b=0x000007FF \text{ e } I_c=0x00000222)$

4.6. Conclusioni

La principale conclusione di questo capitolo è che sia i risultati delle simulazioni che delle prove sperimentali hanno convalidato il metodo proposto nel capitolo 2 per la stima della posizione e velocità angolare. Si dimostra però che utilizzando il DSP si possono avere risultati sperimentali in buon accordo con i risultati delle simulazioni, questo si può verificare facilmente al fare il confronto tra la grafica della posizione angolare ideale e stimata a bassa velocità ottenuta con Simulink (figura 4.2) e quella ottenuta con gli stessi parametri attraverso il DSP (figura 4.22). Un altro confronto si può fare tra le figure 4.19 e 4.36, nelle quali si riportano le componenti del vettore unitario rotante, cioè il $\sin(\theta_{\text{HALL}})$ e $\cos(\theta_{\text{HALL}})$, dove si può visualizzare che con il DSP si ottengono sinusoidi più accurati. In genere, tutti i risultati ottenuti attraverso il DSP sono altamente precisi rispetto agli ottenuti con Simulink.

Purtroppo non sono stati verificati sperimentalmente con il DSP i miglioramenti delle prestazioni del VTO quando si usano i metodi proposti nei paragrafi 2.4.2 e 2.4.3 (Gain Scheduling e disaccoppiamento delle armoniche). Tuttavia, le simulazioni fatte su Simulink (Matlab) mostrano che si migliorano notevolmente le prestazioni del VTO.

Conclusioni Generali

Nel **capitolo 1**, è stato fatto un riassunto sugli azionamenti elettrici, descrivendo lo sviluppo e l'importanza degli azionamenti AC. Tuttavia, in questi azionamenti la necessità di misurare la posizione angolare ad alta risoluzione richiede l'adozione di costosi encoder ad alta risoluzione che fanno aumentare il costo di installazione e manutenzione dell'azionamento. Lo scopo di questa tesi è l'eliminazione di questi costosi encoder ad alta risoluzione; per questo motivo è stata adottata una soluzione a bassa risoluzione attraverso un encoder composto da 2 sonde binarie ad effetto Hall.

Gli algoritmi per la stima della posizione e velocità angolare che vengono utilizzati insieme a questi encoder possono essere classificati in due gruppi: il primo basato sull'extrapolazione ed il secondo basato sul rilevamento della forza elettromotrice. Il primo gruppo è caratterizzato da un ritardo nella stima della velocità ed il secondo dell'avere una dipendenza dei parametri elettrici ed una scarsa prestazione alle basse velocità, poiché in questa regione la forza elettromotrice non è facilmente rilevabile.

Nel **capitolo 2**, è stato dimostrato che la misura ottenuta da 2 sonde ad effetto Hall, sfasate di 90° elettrici intorno al traferro, può essere trattate per formare un vettore spaziale quasi-rotante, denominato Vettore Hall. Si è proposto un algoritmo per la stima della posizione e velocità angolare attraverso un osservatore ad inseguimento vettoriale (Vector Tracking Observer, VTO). E' stata introdotta la linearizzazione intorno al punto di lavoro, e poi l'analisi in tempo discreto del VTO per l'inseguimento del Vettore Hall con la quale si può ottenere la larghezza di banda dell'osservatore. I problemi di quantizzazione che si verificano quando è utilizzato il VTO per inseguire il vettore Hall sono stati descritti.

E' stato anche dimostrato che la velocità con cui l'osservatore insegue il vettore Hall dipende dalle caratteristiche di filtraggio e questa proprietà è stata sfruttata per miglioramenti dell'inseguimento attraverso il Gain Scheduling (i guadagni dipendono dalla velocità). Inoltre, sono stati descritte diverse strategie di disaccoppiamento armonico. Impiegando contemporaneamente il Gain Scheduling ed il disaccoppiamento delle armoniche, si è cercato di inseguire effettivamente la posizione e velocità angolare con basso errore di stima.

La struttura del software per il controllo di posizione del motore ed il codice che viene utilizzato nel DSP è stato presentato nel **capitolo 3**. All'interno di questo software si trovano varie applicazioni; tra queste la più importante è quella riguardante il VTO. Importanti caratteristiche di implementazione includono l'uso di aritmetica a 32-bits per migliorare la precisione nei calcoli nel VTO; con aiuto in alcuni casi delle librerie Qmath (PWM) ed IQMATH (VTO). Si genera una funzione che determina la posizione angolare di riferimento con cui si confronta la posizione angolare stimata ottenuta dall'osservatore. Per ultimo, le variabili del VTO sono visualizzate nell'oscilloscopio attraverso l'utilizzo di un Convertitore Digitale-Analogico.

Infine nel **capitolo 4**, sia i risultati delle simulazioni che delle prove sperimentali hanno convalidato il metodo proposto nel capitolo 2 per la stima della posizione e velocità angolare è effettivo. I risultati sperimentali si sono mostrati in buon accordo con i risultati delle simulazioni. In questo capitolo, sono state eseguite altre due prove per dimostrare la funzionalità del DSP, queste prove sono: la prima è la generazione del segnale modulante ad una frequenza di 15kHz con la quale si può comandare l'inverter 3-phase e la seconda per trovare le correnti I_d e I_q nel sistema di riferimento arbitrario $qd0$, partendo dalle correnti di fase d'ingresso al motore I_a , I_b e I_c con l'intenzione di fare il controllo di corrente della macchina.

Appendice

A.1. DSP281x_Constant.h

```

#####
// FILE:      DSP281x_Constant.h
// TITLE:      DSP281x Constant Definitions.
#####

#ifndef DSP281x_CONSTANT_H
#define DSP281x_CONSTANT_H

#ifdef __cplusplus
extern "C" {
#endif

//-----
/* Costanti per il calcolo della posizione angolare */
//-----

#define DT 6.6666666666666666666666666666667e-5
/* DT = 1/(15kHz) */
#define PI 3.1415926535897932384626433832795
/* Pigreco */
#define Dtheta_hex 0x000022B3
/* Delta theta (in formato 9.23) è pari alla velocità angolare (100 rad/s) in formato 9.7 (diviso
tra pigreco) per il DT(1/15KHz) in formato 0.32 (diviso tra 2) */
#define Dtheta_hex1 0x00004566
/* Delta theta1 (in formato 9.23) è pari alla velocità angolare (100 rad/s) in formato 9.7 (diviso
tra pigreco) per il DT(1/15KHz) in formato 0.32 */
#define Dtheta_hex2 0x00011599
/* Delta theta2 (in formato 9.23) è pari alla velocità angolare (400 rad/s) in formato 9.7 (diviso
tra pigreco) per il DT(1/15KHz) in formato 0.32 */
#define Dtheta_hex3 0x0116
/* Delta theta3 (in formato Q.15(int16)) è pari alla velocità angolare (400 rad/s) diviso tra
pigreco) per il DT(1/15KHz) */
#define Dtheta_hex3f_400 8.4882636315677512410071340465341e-3
#define Dtheta_hex3f_200 4.244131815783875620503567023267e-3
#define Dtheta_hex3f_10 2.1220659078919378102517835116335e-4
/* Deltatheta 3 per generare la posizione proveniente da velocità costante pari a 10rad/s,
200rad/s e 400rad/s in formato float */
#define _2div3_ 0x5555
/* 2 su 3 in formato Q.15(int16) */
#define _1div3_ 0x2AAA
/* 1 su 3 in formato Q.15(int16) */
#define _inv_rad_3_ 0x49E6
/* l'inverso della radice di 3 in formato Q.15(int16) */
#define _inv_rad_3_neg_ 0xB61A
/* l'inverso della radice di 3 in formato Q.15(int16) */
#define uno 1

```

```

#define menouno -1
#define due 2

//-----
/* Costanti K1, K2 e K3divK2, necessarie per gli anelli derivativo-proporzionale-integrale, con
Ts=1/5kHz */
//-----
#define k_1 0.0066
#define k_2 7.8991e-6
#define k_3divk_2 1.1869e-4

//-----
/* Costante KI, utilizzata nel torque feedforward */
//-----
#define KI 0
/* Nullo perché il segnale di feedforward non ha alcuna influenza sul posizionamento dei poli
dell'anello chiuso dell'osservatore */
#define I_Ucfft 0
/* Corrente che produce detta coppia */

#ifdef __cplusplus
}
#endif /* extern "C" */

#endif // end of DSP281x_CONSTANT_H definition

//=====
//                      END OF DEFINITIONS
//=====

```

A.2. DSP281x_Device.h

```

#####
// FILE:  DSP281x_Device.h
// TITLE: DSP281x Device Definitions.
#####

#ifndef DSP281x_DEVICE_H
#define DSP281x_DEVICE_H

#ifdef __cplusplus
extern "C" {
#endif

#define TARGET 1
//-----
// User To Select Target Device:

#define DSP28_F2812 TARGET
#define DSP28_F2811 0
#define DSP28_F2810 0

//-----

```

// Common CPU Definitions:

```
extern cregister volatile unsigned int IFR;
extern cregister volatile unsigned int IER;
```

```
#define EINT  asm(" clrc INTM")
#define DINT  asm(" setc INTM")
#define ERTM  asm(" clrc DBGm")
#define DRTM  asm(" setc DBGm")
#define EALLOW asm(" EALLOW")
#define EDIS  asm(" EDIS")
#define ESTOP0 asm(" ESTOP0")
```

```
#define M_INT1  0x0001
#define M_INT2  0x0002
#define M_INT3  0x0004
#define M_INT4  0x0008
#define M_INT5  0x0010
#define M_INT6  0x0020
#define M_INT7  0x0040
#define M_INT8  0x0080
#define M_INT9  0x0100
#define M_INT10 0x0200
#define M_INT11 0x0400
#define M_INT12 0x0800
#define M_INT13 0x1000
#define M_INT14 0x2000
#define M_DLOG  0x4000
#define M_RTOS  0x8000
```

```
#define BIT0  0x0001
#define BIT1  0x0002
#define BIT2  0x0004
#define BIT3  0x0008
#define BIT4  0x0010
#define BIT5  0x0020
#define BIT6  0x0040
#define BIT7  0x0080
#define BIT8  0x0100
#define BIT9  0x0200
#define BIT10 0x0400
#define BIT11 0x0800
#define BIT12 0x1000
#define BIT13 0x2000
#define BIT14 0x4000
#define BIT15 0x8000
```

```
//-----
// For Portability, User Is Recommended To Use Following Data Type Size
// Definitions For 16-bit and 32-Bit Signed/Unsigned Integers:
```

```
#ifndef DSP28_DATA_TYPES
#define DSP28_DATA_TYPES
typedef int          int16;
```

```

typedef long          int32;
typedef unsigned int  Uint16;
typedef unsigned long Uint32;
typedef float         float32;
typedef long double   float64;
#endif

//-----
// Include All Peripheral Header Files:

#include "DSP281x_SysCtrl.h"    // System Control/Power Modes
#include "DSP281x_DevEmu.h"    // Device Emulation Registers
#include "DSP281x_Xintf.h"     // External Interface Registers
#include "DSP281x_CpuTimers.h" // 32-bit CPU Timers
#include "DSP281x_PieCtrl.h"   // PIE Control Registers
#include "DSP281x_PieVect.h"   // PIE Vector Table
#include "DSP281x_Spi.h"       // SPI Registers
#include "DSP281x_Sci.h"       // SCI Registers
#include "DSP281x_Mcbsp.h"     // McBSP Registers
#include "DSP281x_ECan.h"      // Enhanced eCAN Registers
#include "DSP281x_Gpio.h"      // General Purpose I/O Registers
#include "DSP281x_Ev.h"        // Event Manager Registers
#include "DSP281x_Adc.h"       // ADC Registers
#include "DSP281x_XIntrupt.h"  // External Interrupts

#ifdef __cplusplus
}
#endif /* extern "C" */

#endif // end of DSP281x_DEVICE_H definition

//=====
// No more.
//=====

```

A.3. DSP281x_Examples.h

```

#####
// FILE:  DSP281x_Examples.h
// TITLE: DSP281x Device Definitions.
#####

#ifndef DSP281x_EXAMPLES_H
#define DSP281x_EXAMPLES_H

#ifdef __cplusplus
extern "C" {
#endif

/*-----
Specify the clock rate of the CPU (SYSCLKOUT) in nS.

Take into account the input clock frequency and the PLL multiplier

```

selected in SysCtrl.c (0xA by default).

Use one of the values provided, or define your own.
The trailing L is required tells the compiler to treat the number as a 64-bit value.

Only one statement should be uncommented.

Example: CLKIN is a 30MHz crystal.

In SysCtrl.c the user specified PLLCR = 0xA for a 150Mhz CPU clock (SYSCLKOUT = 150MHz).

In this case, the CPU_RATE will be 6.667L
Uncomment the line: #define CPU_RATE 6.667L

```
-----*/

#define CPU_RATE 6.667L // for a 150MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 7.143L // for a 140MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 8.333L // for a 120MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 10.000L // for a 100MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 13.330L // for a 75MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 20.000L // for a 50MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 33.333L // for a 30MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 41.667L // for a 24MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 50.000L // for a 20MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 66.667L // for a 15MHz CPU clock speed (SYSCLKOUT)
// #define CPU_RATE 100.000L // for a 10MHz CPU clock speed (SYSCLKOUT)
//-----
// Include Example Header Files:

#include "DSP281x_GlobalPrototypes.h" // Prototypes for global functions within the
// .c files.

#include "DSP281x_SWPrioritizedIsrLevels.h" // Used for Software Prioritization of ISR's

// Include files not used with DSP/BIOS
#ifndef DSP28_BIOS
#include "DSP281x_DefaultISR.h"
#endif

// DO NOT MODIFY THIS LINE.
#define DELAY_US(A) DSP28x_usDelay((((long double) A * 1000.0L) / (long double)CPU_RATE) - 9.0L) / 5.0L)

#ifdef __cplusplus
}
#endif /* extern "C" */

#endif // end of DSP281x_EXAMPLES_H definition

//=====
// No more.
//=====
```

A.4. DSP281x_AD7568.h

```
#####
// FILE: DSP281x_AD7568.h
// TITLE: DAC AD7568 Definitions.
#####

#ifndef AD7568_H
#define AD7568_H

#ifdef __cplusplus
extern "C" {
#endif

extern int16 DAC_buffer[8];

extern int16 Ad7568ChAddr[8];

#ifdef __cplusplus
}
#endif /* extern "C" */

#endif // end of DSP281x_AD7568.h definition

//=====
// No more.
//=====
```

A.5. DSP281x_Osservatore.c

```
#####
// APPLICAZIONE: "Sviluppo di algoritmi di controllo per un azionamento AC Brushless con
//               sonde ad effetto Hall digitali"
//
// FILE:         DSP281x_Osservatore.c
//
// TITOLO:       Vector Tracking Observer VTO (Osservatore ad inseguimento vettoriale)
//
// AUTORE:       Manuel Alejandro Martinez Vaamonde
//
// DATA:       Aprile 2008
//
// DESCRIZIONE:  Lo scopo di questo programma è realizzare un osservatore ad
//               inseguimento vettoriale attraverso il segnale di 2 sonde ad effetto Hall
//               messe in quadratura sugli assi q e d, con cui si può ottenere la posizione
//               angolare per fare i rispettivi controlli di velocità, corrente e coppia di un
//               motore sincrono.
//
// "La Sapienza" - Università di Roma
// Dipartimento di Ingegneria Elettrica
#####
```

APPENDICE

```

#####
//          INCLUDE GENERAL SYSTEM PARAMETERS E LIBRERIES
#####

#include "DSP281x_Device.h"    /* DSP281x Headerfile Include File */
#include "DSP281x_Examples.h"  /* DSP281x Examples Include File */
#include "qmath.h"             /* Math Library */
#include "DSP281x_Constant.h"  /* DSP281x Constant Include File */
#include "DSP281x_AD7568.h"    /* MAZEL Software Include File */
#include "IQmathLib.h"         /* IQmath Library */

#####
//          PROTOTYPE STATEMENTS FOR FUNCTION
#####

void init_eva(void);
interrupt void motorcontrol(void);
void spi_fifo_init(void);
void AD7568_init(void);
void AD7568_write(int16 buf[8]);

#####
//          VARIABLES
#####

/* Integers (16-bits) */
int posiz1,posiz2,posiz3;
int g1,g2,g3,g1_1,g2_2,g3_3,vts_int16;
int Ia,Ib,Ic,I_alpha,I_beta,I_d_mot,I_q_mot;
int Ib_x_inv_rad_3_,Ic_x_inv_rad_3_;
int Ad7568ChAddr[8],adc_input,DAC_buffer[8];
int Halpha_K_int16,Hbeta_K_int16;
/* Long integer (32-bits) */
long Sin1,Cos1,Sin2,Sin3,I_d,I_q;
long prod1,prod2,prod3,e1,e2,e3;
long prod1_shiftato,prod2_shiftato,prod3_shiftato;
long Pr_alpha1,Pr_alpha2,Pr_beta1,Pr_beta2;
long e1_shiftato,e2_shiftato,e3_shiftato;
/* Unsigned integer 16-bits */
unsigned int r1,r2,r3;
/* Unsigned integer 32-bits */
unsigned long MisuraestPWM1,MisuraestPWM2,MisuraestPWM3;
unsigned long Misura_Ia,Misura_Ib,Misura_Ic;
unsigned long p1,p2,p3,q1,q2,q3;
/* _iq .....Fixed point datatype: GLOBAL_Q (=29) format */
_iq uno_iq29,menouno_iq29,due_iq29,PI_iq29,menoPI_iq29;
_iq Dtheta_prov_10,Dtheta_prov_200,Dtheta_prov_400,posiz_prov;
_iq m1,m2,m3,m4,Ha,Hb,i1;
_iq sin_theta_hall,cos_theta_hall,sin_theta_hall_div,cos_theta_hall_div;
_iq k_1_iq29,k_2_iq29,k_3divk_2_iq29,KI_iq29,I_Ucfft_iq29,a;
_iq error_alpha,error_beta,obs_error,obs_error_1,K2_obs_error,K2_obs_error_s1;
_iq int_plus_prop,acc_obs_error,acc_obs_error1,torque_feedforward,K1_obs_error;
_iq omega_dot_hat,omega_dot_hat1,omega_hat,omega_hat1;
_iq omega_hat_enh,omega_hat_enh1,theta_hall,theta_hall_pu;

```



```

/* _iq15 .....Fixed point datatype: Q15 format */
_iq15 posiz_prov_vis,Halpha,Hbeta,sin_theta_hall_vis,cos_theta_hall_vis;
_iq15 error_alpha_vis,error_beta_vis,obs_error_vis,K2_obs_error_vis;
_iq15 acc_obs_error_vis,a_vis,omega_dot_hat_vis,omega_hat_vis,omega_hat_enh_vis;
_iq15 theta_hall_int16,error_theta,error_theta_m,error_theta_vis;

#####
//                                START OF PROGRAM CODE
#####

void main(void)
{
/* Initialize posiz1, posiz2, posiz3 */
posiz1 = 0x0000;
posiz2 = 0x0000;
posiz3 = 0x0000;
posiz_prov = 0x0000;
i1 = 0x0000;
acc_obs_error = 0x0000;
theta_hall = 0x0000;
omega_hat = 0x0000;
obs_error = 0x0000;

/* Step 1. Initialize System Control:
PLL, WatchDog, enable Peripheral Clocks
This function is found in the DSP281x_SysCtrl.c file.*/
InitSysCtrl();

/* Step 2. Initialize GPIO:
This function is found in the DSP281x_Gpio.c file and
illustrates how to set the GPIO to it's default state.
InitGpio(); Skipped for this program */

/* For this program, set HSPCLK to SYSCLKOUT / 6
(25Mhz assuming 150Mhz SYSCLKOUT) */
EALLOW;
SysCtrlRegs.HISPCP.all = 0x0003; // HSPCLK = SYSCLKOUT/6
EDIS;

/* Initialize only GPAMUX, GPBMUX, GPBDIR and GPFMUX for this program */
EALLOW;
// Enable PWM pins
GpioMuxRegs.GPAMUX.all = 0x00FF;    // EVA PWM 1-6 pins

// Enable GPIO pins
GpioMuxRegs.GPBMUX.bit.PWM7_GPIOB0 = 0; // Porta Digitale 1 (PD1) attiva
GpioMuxRegs.GPBMUX.bit.PWM8_GPIOB1 = 0; // Porta Digitale 2 (PD2) attiva
GpioMuxRegs.GPBMUX.bit.PWM9_GPIOB2 = 0; // Porta Digitale 3 (PD3) attiva
GpioMuxRegs.GPBMUX.bit.PWM10_GPIOB3 = 0; // Porta Digitale 4 (PD4) attiva
GpioMuxRegs.GPBMUX.bit.PWM11_GPIOB4 = 0; // Porta Digitale 5 (PD5) attiva
GpioMuxRegs.GPBMUX.bit.PWM12_GPIOB5 = 0; // Porta Digitale 6 (PD6) attiva
GpioMuxRegs.GPBDIR.bit.GPIOB0 = 1;    // Impostazione PD1 in uscita
GpioMuxRegs.GPBDIR.bit.GPIOB1 = 1;    // Impostazione PD2 in uscita
GpioMuxRegs.GPBDIR.bit.GPIOB2 = 1;    // Impostazione PD3 in uscita

```

```

GpioMuxRegs.GPBDIR.bit.GPIOB3 = 1;    // Impostazione PD4 in uscita
GpioMuxRegs.GPBDIR.bit.GPIOB4 = 1;    // Impostazione PD5 in uscita
GpioMuxRegs.GPBDIR.bit.GPIOB5 = 1;    // Impostazione PD6 in uscita
// Enable SPI functionality
GpioMuxRegs.GPFMUX.all=0x000D; // Select GPIOs to be SPI pins except for
                                // SPISOMI Port F MUX - x000 0000 0000 1101

// Enable Digital outputs
GpioMuxRegs.GPFDIR.bit.GPIOF1 = 1; // SPISOMI is used as digital output
GpioMuxRegs.GPFDIR.bit.GPIOF6 = 1; // CANTX is used as digital output
EDIS;

/* Step 3. Clear all interrupts and initialize PIE vector table:
Disable CPU interrupts */
DINT;

/* Initialize PIE control registers to their default state.
The default state is all PIE interrupts disabled and flags
are cleared.
This function is found in the DSP281x_PieCtrl.c file. */
InitPieCtrl();

/* Disable CPU interrupts and clear all CPU interrupt flags: */
IER = 0x0000;
IFR = 0x0000;

/* Initialize the PIE vector table with pointers to the shell Interrupt
Service Routines (ISR).
This will populate the entire table, even if the interrupt
is not used in this program. This is useful for debug purposes.
The shell ISR routines are found in DSP281x_DefaultIsr.c.
This function is found in DSP281x_PieVect.c. */
InitPieVectTable();

/* Interrupts that are used here are re-mapped to ISR functions found
within this file. */
EALLOW; // This is needed to write to EALLOW protected register
PieVectTable.T1PINT = &motorcontrol;
EDIS; // This is needed to disable write to EALLOW protected registers

/* Step 4. Initialize all the Device Peripherals:
This function is found in DSP281x_InitPeripherals.c
InitPeripherals(); Not required for this example */
init_eva();
InitAdc();
spi_fifo_init(); // Initialize the SPI
AD7568_init();

/* Step 5. User specific code, enable interrupts: */

/* Enable EVA */
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE block
PieCtrlRegs.PIEIER2.bit.INTx4 = 1; // PIE, INT2 Group Enable Register
IER |= M_INT2;                      // Enable CPU Interrupt 2
EINT;                               // Enable Global Interrupt INTM

```

```

    ERTM; // Enable Global Realtime Interrupt DBG
    EvaRegs.EVAIMRA.bit.T1PINT = 1; // GP timer 1 period interrupt
    // EVA Interrupt Mask Register A

/* Configure ADC */
    AdcRegs.ADCMAXCONV.all = 0x0005; // Setup 4 conv's on SEQ1 and
    // 2 conv's on SEQ2
    AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0000; // Setup conv from ADCINA0 &
    // ADCINB0
    AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x0001; // Setup conv from ADCINA1 &
    // ADCINB1
    AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x0002; // Setup conv from ADCINA2 &
    // ADCINB2
    AdcRegs.ADCTRL2.bit.EVA_SOC_SEQ1 = 1; // Enable EVASOC to start SEQ1
    AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 0; // Enable SEQ1 interrupt
    // (every EOS)

/* Just sit and loop forever: PWM pins can be observed with a scope. */
    for(;;);

}

void init_eva()
{

/* EVA Configure T1PWM, T2PWM, PWM1-PWM6 */
/* Initialize the timers */

/* Initialize EVA Timer1 */
    EvaRegs.T1PR = 0x0682; // Setup Timer1 period register,must to be
    // 0x1388 (75MHz/15kHz) but it's setup as
    // 0x1388/3 because ADConverter works at
    // 25 MHz
    EvaRegs.T1CMPR = 0x3C00; // Setup Timer1 compare value
    EvaRegs.T1CNT = 0x0000; // Timer1 counter

    //TMODE = continuous up/down
    //Timer enable
    //Timer compare enable

    EvaRegs.GPTCONA.bit.T1TOADC = 1; // Enable EVASOC in EVA
    EvaRegs.T1CON.all = 0x1042; // Enable Timer 1 Compare (upcount mode)

/* Setup T1PWM and T2PWM */
/* Drive T1/T2 PWM by compare logic */
    EvaRegs.GPTCONA.bit.TCMPOE = 1;
/* Polarity of GP Timer 1 Compare = Active low */
    EvaRegs.GPTCONA.bit.T1PIN = 1;

/* Enable compare for PWM1-PWM6 */
    EvaRegs.CMPR1 = 0x0C00;
    EvaRegs.CMPR2 = 0x3C00;
    EvaRegs.CMPR3 = 0xFC00;

```

```

/* Compare action control. Action that takes place on a compare event */
//output pin 1 CMPR1 - active high
//output pin 2 CMPR1 - active low
//output pin 3 CMPR2 - active high
//output pin 4 CMPR2 - active low
//output pin 5 CMPR3 - active high
//output pin 6 CMPR3 - active low

EvaRegs.ACTRA.all = 0x0666;
EvaRegs.DBTCNA.all = 0x0000; // Disable deadband
EvaRegs.COMCONA.all = 0xA600;

}

void spi_fifo_init(void)
{
// Initialize SPI FIFO registers

SpiaRegs.SPICCR.bit.SPISWRESET=0; // Reset SPI

SpiaRegs.SPICCR.all=0x000F; // 16-bit character,data is output on rising edge
SpiaRegs.SPICTL.all=0x0017; // Interrupt enabled, Master/Slave XMIT enabled
SpiaRegs.SPISTS.all=0x0000;
SpiaRegs.SPIBRR=0x0003; // Baud rate
SpiaRegs.SPIFFTX.all=0xC028; // Enable FIFO's, set TX FIFO level to 8
SpiaRegs.SPIFFRX.all=0x0000; // Disable RX FIFO Interrupt because SPISOMI is used
                             as
                             //GPIO pin (digital output)

SpiaRegs.SPIFFCT.all=0x01;
SpiaRegs.SPIPRI.all=0x0010;

SpiaRegs.SPICCR.bit.SPISWRESET=1; // Enable SPI

SpiaRegs.SPIFFTX.bit.TXFIFO=1;
SpiaRegs.SPIFFRX.bit.RXFIFORESET=1;

}

void AD7568_init(void)
{
    Ad7568ChAddr[0]=0; // DAC channel 1 address
    Ad7568ChAddr[1]=1; // DAC channel 2 address
    Ad7568ChAddr[2]=2; // DAC channel 3 address
    Ad7568ChAddr[3]=3; // DAC channel 4 address
    Ad7568ChAddr[4]=4; // DAC channel 5 address
    Ad7568ChAddr[5]=5; // DAC channel 6 address
    Ad7568ChAddr[6]=6; // DAC channel 7 address
    Ad7568ChAddr[7]=7; // DAC channel 8 address

}

```

```
void AD7568_write(int16 buf[8])
```

```
{
    Uint16 i;
    Uint16 send_buf[8];

    for(i=0;i<8;i++)
    {
        send_buf[i] = buf[i]&0xFFF0;           // remove LSB nibble
        send_buf[i] = Ad7568ChAddr[i]|send_buf[i]; // add DAC address
        send_buf[i] = 0x8000 + send_buf[i];     // convert from signed to unsigned
integer
    }

    for(i=0;i<8;i++)
    {
        SpiaRegs.SPITXBUF=send_buf[i];    // Send data
    }
}
```

```
interrupt void motorcontrol(void)
```

```
{

    GpioDataRegs.GPBSET.bit.GPIOB5 = 1;
    /* Porta digitale 6, impostata in alto per misurare il tempo che si mette la routine per
    effettuare le operazione di PWM, ABC -> ALPHA,BETA -> DQ, VTO e DAC */

    //-----
    /*##### PULSE-WIDTH MODULATION (PWM) #####*/
    //-----

    GpioDataRegs.GPBSET.bit.GPIOB0 = 1;
    /*Porta digitale 2, impostata in alto per misurare il tempo di esecuzione del PWM includendo
    la generazione della posizione angolare ideale del rotore del motore */

    //-----
    /*##### Posizione angolare ideale del Rotore del Motore #####*/
    //-----

    while(AdcRegs.ADCST.bit.SEQ1_BSY == 1 )
    { asm (" NOP"); }

    vts_int16 = Dtheta_hex3;
    /* Si dichiara in formato Q.15(int16), per poterla visualizzare attraverso il Digital-Analogic
    Converter (DAC) */

    posiz1 = posiz1 + vts_int16;
    /* Posizione angolare ideale del rotore del motore, utilizzata come riferimento */

    posiz2 = posiz1 + _2div3_;
    /* Posizione angolare ideale del rotore del motore spostata in +120gradi elettrici */

    posiz3 = posiz1 - _2div3_;
    /* Posizione angolare ideale del rotore del motore spostata in -120gradi elettrici */
}
```

```
//-----
/***** Struttura della PWM *****/
//-----

Sin1 = qsin(posiz1);
Cos1 = qcos(posiz1);
Sin2 = qsin(posiz2);
Sin3 = qsin(posiz3);
/* PWM Sinusoidale */

//-----
/***** Utilizzando l'ADC *****/
//-----

/* MisuraestPWM1 = AdcRegs.ADCRESULT0 >> 4;
MisuraestPWM2 = AdcRegs.ADCRESULT1 >> 4;
MisuraestPWM3 = AdcRegs.ADCRESULT2 >> 4;*/
/* Acquisizione analogica (Potenziometro) per la conversione a digitale, si tolgono i primi 4
bits per condizioni del ADC. Con questo potenziometro riesco ad ottenere un duty cycle
variabile, cioè da libera impostazione */

/* prod1 = Sin1 * MisuraestPWM1;
prod2 = Sin2 * MisuraestPWM2;
prod3 = Sin3 * MisuraestPWM3;*/
/* Prodotto tra il valore Misurato per il ADC ed il seno (l'argomento del seno sarà la posizione
angolare ideale di riferimento) */

/* prod1_shiftato = prod1 >> 12;
prod2_shiftato = prod2 >> 12;
prod3_shiftato = prod3 >> 12;*/
/* Si tolgono i 12 bits meno significativi per portarlo al formato 1.15 */

/* p1 = prod1_shiftato + 0x8000;
p2 = prod2_shiftato + 0x8000;
p3 = prod3_shiftato + 0x8000;*/
/* Si porta al formato senza segno hexadecimal 16.0 */

/* q1 = p1 * (0x0682);
q2 = p2 * (0x0682);
q3 = p3 * (0x0682);*/
/* Si prescala per ottenere come ampiezza massima della modulante (sinusoidale) il valore in
hexadecimal di 0x0682 */

/* r1 = q1 / (0xFFFF);
r2 = q2 / (0xFFFF);
r3 = q3 / (0xFFFF);*/
/* Si fa un shift di 16 bits alla sinistra per diventarlo formato 0.16 */

/* EvaRegs.CMPR1 = r1;
EvaRegs.CMPR2 = r2;
EvaRegs.CMPR3 = r3;*/
/* Uscita della PWM Sinusoidale alla frequenza desiderata f =
(400rad/s)/(2*pi greco)=63,66197Hz */
```

```
//-----
/***** Senza utilizzare l'ADC *****/
//-----

e1 = Sin1 * 0x0682;
e2 = Sin2 * 0x0682;
e3 = Sin3 * 0x0682;
/* Il prodotto mi prescala l'onda per ottenere come ampiezza massima della modulante
(sinusoidale) il valore in hexadecimale di 0x0682 */

e1_shiftato = e1 >> 16;
e2_shiftato = e2 >> 16;
e3_shiftato = e3 >> 16;
/* Si prende i 16 bits più significativi dal prodotto precedente */

g1 = e1_shiftato;
g2 = e2_shiftato;
g3 = e3_shiftato;
/* Si riporta in formato Q.15(int16) per poter visualizzare il segnale attraverso il DAC */

g1_1 = g1 + 0x0341;
g2_2 = g2 + 0x0341;
g3_3 = g3 + 0x0341;
/* Si aggiunge l'offset, pari alla ampiezza massima della modulante (0x0682) diviso 2 */

EvaRegs.CMPR1 = g1_1;
EvaRegs.CMPR2 = g2_2;
EvaRegs.CMPR3 = g3_3;
/* Uscita della PWM Sinusoidale alla frequenza desiderata f =
(400rad/s)/(2*pi greco)=63,66197Hz */

EvaRegs.EVAIFRA.bit.T1PINT = 1; // GP timer 1 period interrupt
// EVA Interrupt Flag Register A

/* Reinitialize for next ADC sequence */
/* Reset SEQ1 */
AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;

/* To receive more interrupts from this PIE group, acknowledge this interrupt */
PieCtrlRegs.PIEACK.all = PIEACK_GROUP2;

GpioDataRegs.GPBCLEAR.bit.GPIOB0 = 1;
/* Porta digitale 1, impostata in basso sempre per calcolare il tempo di esecuzione del PWM
includendo la generazione della posizione angolare ideale del rotore del motore */

//-----
/***** ABC -> ALPHA,BETA -> DQ COORDINATES TRANSFORMATION *****/
//-----

GpioDataRegs.GPBSET.bit.GPIOB1 = 1;
/*Porta digitale 2, impostata in alto per misurare il tempo di esecuzione del ABC ->
ALPHA,BETA -> DQ COORDINATES TRANSFORMATION */
```

```
//-----
/***** Utilizzando l'ADC *****/
//-----

/* Misura_Ia = AdcRegs.ADCRESULT3 >> 4;
   Misura_Ib = AdcRegs.ADCRESULT4 >> 4;
   Misura_Ic = AdcRegs.ADCRESULT5 >> 4;*/
/* Acquisizione analogica delle correnti di fase (A,B,C) d'ingresso al motore prese con sonde
ad effetto Hall di corrente per la conversione a digitale, si tolgono i primi 4 bits per condizioni
del ADC */

/* Ia = Misura_Ia - 0x0800;
   Ib = Misura_Ib - 0x0800;
   Ic = Misura_Ic - 0x0800;*/
/* Correnti di fase A, B e C d'ingresso al motore in formato Q.15(int16),prese con sonde ad
effetto Hall di Corrente */

/* Ib_x_inv_rad_3_ = Ib * _inv_rad_3_;
   Ic_x_inv_rad_3_ = Ic * _inv_rad_3_;*/
/* Prodotto tra le correnti di fase b,c e l'inverso di radice di 3,necessario per il cambio di
sistema di coordinate */

/* I_alpha = Ia;
   I_beta = Ib_x_inv_rad_3_ - Ic_x_inv_rad_3_;*/
/* Correnti nel sistema di riferimento stazionario alpha e beta*/

/* Pr_alpha1 = Sin1 * I_alpha;
   Pr_alpha2 = Cos1 * I_alpha;
   Pr_beta1 = Sin1 * I_beta;
   Pr_beta2 = Cos1 * I_beta;*/
/* Prodotto necessario per il cambio di sistema di coordinate */

/* Id = Pr_alpha1 - Pr_beta2;
   Iq = Pr_alpha2 + Pr_beta1;*/
/* Correnti nel sistema di riferimento arbitrario qd0 */

/* Id_mot = Id >> 16;
   Iq_mot = Iq >> 16;*/
/* Correnti in formato Q.15(int16) nel sistema di riferimento arbitrario qd */

//-----
/***** Senza utilizzare l'ADC *****/
//-----

Misura_Ia = 0x00000FFF;
Misura_Ib = 0x000007FF;
Misura_Ic = 0x00000000;
/* Correnti di fase (A,B,C) impostata per prova della trasformata */

Ia = Misura_Ia - 0x0800;
Ib = Misura_Ib - 0x0800;
Ic = Misura_Ic - 0x0800;
/* Correnti di fase A, B e C d'ingresso al motore in formato Q.15(int16),prese
con sonde ad effetto Hall di Corrente */
```



```

    lb_x_inv_rad_3_ = lb * _inv_rad_3_;
    lc_x_inv_rad_3_ = lc * _inv_rad_3_;
/* Prodotto tra le correnti di fase b,c e l'inverso di radice di 3,necessario per
il cambio di sistema di coordinate */

    I_alpha = Ia;
    I_beta = lb_x_inv_rad_3_ - lc_x_inv_rad_3_;
/* Correnti nel sistema di riferimento stazionario alpha e beta*/

    Pr_alpha1 = Sin1 * I_alpha;
    Pr_alpha2 = Cos1 * I_alpha;
    Pr_beta1 = Sin1 * I_beta;
    Pr_beta2 = Cos1 * I_beta;
/* Prodotto necessario per il cambio di sistema di coordinate */

    Id = Pr_alpha1 - Pr_beta2;
    Iq = Pr_alpha2 + Pr_beta1;
/* Correnti nel sistema di riferimento arbitrario qd0 */

    Id_mot = Id >> 16;
    Iq_mot = Iq >> 16;
/* Correnti in formato Q.15(int16) nel sistema di riferimento arbitrario qd */

    GpioDataRegs.GPBCLEAR.bit.GPIOB1 = 1;
/* Porta digitale 2, impostata in basso sempre per calcolare il tempo di esecuzione del ABC -
> ALPHA,BETA -> DQ COORDINATES TRANSFORMATION */

//-----
/*##### VECTOR TRACKING OBSERVER VTO #####*/
//-----

    GpioDataRegs.GPBSET.bit.GPIOB2 = 1;
/*Porta digitale 3, impostata in alto per misurare il tempo di esecuzione dell'osservatore
includendo posizione angolare di riferimento, Halpha e Hbeta e l'osservatore ad
inseguimento vettoriale */

    uno_iq29 = _IQ(uno);
    menouno_iq29 = _IQ(menouno);
    due_iq29 = _IQ(due);
    PI_iq29 = _IQ(PI);
    menoPI_iq29 = _IQ(-PI);

//-----
/****** Posizione angolare ideale del Rotore del Motore *****/
//-----

    // Dtheta_prov_10 = _IQ(Dtheta_hex3f_10);
/* Delta theta per una velocità angolare pari a 10rad/s */
    // Dtheta_prov_200 = _IQ(Dtheta_hex3f_200);
/* Delta theta per una velocità angolare pari a 200rad/s */
    Dtheta_prov_400 = _IQ(Dtheta_hex3f_400);
/* Delta theta per una velocità angolare pari a 400rad/s */

```

```

// posiz_prov = posiz_prov + Dtheta_prov_10;
/* Posizione angolare di riferimento, con Velocità Angolare 10rad/s */
// posiz_prov = posiz_prov + Dtheta_prov_200;
/* Posizione angolare di riferimento, con Velocità Angolare 200rad/s */
posiz_prov = posiz_prov + Dtheta_prov_400;
/* Posizione angolare di riferimento, con Velocità Angolare 400rad/s */

if (posiz_prov >= uno_iq29)
posiz_prov = menouno_iq29;
/* Ciclo creato per impostare la posizione angolare stimata (in per unità) tra -1 e +1 */

posiz_prov_vis = _IQtoIQ15(posiz_prov);
/* posiz_prov in formato Q.15 per condizioni del DAC*/

//-----
/***** Segnali Halpha e Hbeta delle sonde ad effetto Hall *****/
//-----

if ((posiz_prov >= 0xE0000000) & (posiz_prov < 0xF0000000))
/*theta_quant_K1 = (pi/4)*/
/*m1 = 0.25*pi in _iq29 */
{m1 = 0x1921FB54;
Ha = 0x16A09E66; // Ha=_IQcos(m1);
Hb = 0x16A09E66;} // Hb=_IQsin(m1);

else if ((posiz_prov >= 0xF0000000) & (posiz_prov <= 0x00000000))
/*theta_quant_K2 = (3*pi/4)*/
/*m2 = 0.75*pi in _iq29 */
{m2 = 0x4B65F1FC;
Ha = 0xE95F619A; // Ha=_IQcos(m2);
Hb = 0x16A09E66;} // Hb=_IQsin(m2);

else if ((posiz_prov >= 0x00000000) & (posiz_prov < 0x10000000))
/*theta_quant_K3 = (-3*pi/4)*/
/*m3 = -0.75*pi in _iq29 */
{m3 = 0xB49A0E04;
Ha = 0xE95F619A; // Ha=_IQcos(m3);
Hb = 0xE95F619A;} // Hb=_IQsin(m3);

else if ((posiz_prov >= 0x10000000) & (posiz_prov <= 0x20000000))
/*theta_quant_K4 = (-pi/4)*/
/*m4 = -0.25*pi in _iq29 */
{m4 = 0xE6DE04AC;
Ha = 0x16A09E66; // Ha=_IQcos(m4);
Hb = 0xE95F619A;} // Hb=_IQsin(m4);

Halpha = _IQtoIQ15(Ha);
Hbeta = _IQtoIQ15(Hb);
/* Halpha(Ha) e Hbeta(Hb) in formato Q.15 per condizioni del DAC*/

```

```
//-----
/***** Algoritmo interno dell'osservatore *****/
//-----

if (i1==1)
{

GpioDataRegs.GPBSET.bit.GPIOB3 = 1;
/* Porta digitale 4, impostata in alto per misurare il tempo che si mette l'osservatore per
l'inseguimento della posizionene angolare */

sin_theta_hall = _IQsin(theta_hall);
cos_theta_hall = _IQcos(theta_hall);
/* Calcolo del seno e coseno che costituiscono il blocco "vettore unitario",cioè
e^(jtheta_hall)= cos(theta_hall) + j*sin(theta_hall) */

sin_theta_hall_div = _IQdiv(sin_theta_hall,_IQ29(2));
cos_theta_hall_div = _IQdiv(cos_theta_hall,_IQ29(2));
/* sin(theta_hall) e cos(theta_hall) diviso 2, per rendere le due funzioni simmetriche rispetto
allo zero */

sin_theta_hall_vis = _IQtoQ15(sin_theta_hall);
cos_theta_hall_vis = _IQtoQ15(cos_theta_hall);
/* sin(theta_hall) e cos(theta_hall) in formato Q.15 per condizioni del DAC*/

error_alpha = _IQmpyIQX(Ha,29,sin_theta_hall_div,29);
/* Prodotto vettoriale tra Halpha e Sin(theta_hall) */
error_alpha_vis = _IQtoQ15(error_alpha);
/* error_alpha in formato Q.15 per condizioni del DAC*/

error_beta = _IQmpyIQX(Hb,29,cos_theta_hall_div,29);
/* Prodotto vettoriale tra Hbeta e Cos(theta_hall) */
error_beta_vis = _IQtoQ15(error_beta);
/* error_beta in formato Q.15 per condizioni del DAC */

obs_error_1 = error_alpha - error_beta;
/* Differenza tra l'error_alpha e l'error_beta */
obs_error = _IQmpyIQX(obs_error_1,29,due_iq29,29);
/* Moltiplicato per 2, per compensare la divisione tra 2 del sin(theta_hall) e cos(theta_hall)*/
obs_error_vis = _IQtoQ15(obs_error);
/* error_beta in formato Q.15 per condizioni del DAC */

k_2_iq29 = _IQ(k_2);
/* k_2 è definita nel DSP281x_Constant.h */
K2_obs_error = _IQmpyIQX(k_2_iq29,29,obs_error_1,29);
/* K2_obs_error = k_2_iq29 * obs_error_1 */
K2_obs_error_s1 = K2_obs_error << 16;
K2_obs_error_vis = _IQtoQ15(K2_obs_error_s1);
/* K2_obs_error in formato Q.15 per condizioni del DAC */

acc_obs_error = acc_obs_error + K2_obs_error;
/* Prima trasformata z {1/(1-Z^-1)} */
acc_obs_error1 = acc_obs_error <<16;
acc_obs_error_vis = _IQtoQ15(acc_obs_error1);
```

```

/* acc_obs_error in formato Q.15 per condizioni del DAC */

k_3divk_2_iq29 = _IQ(k_3divk_2);
/* k_3divk_2 è definito nel DSP281x_Constant.h */
a = _IQmpyIQX(k_3divk_2_iq29,29,acc_obs_error,29);
/* a = k_3divk_2 * acc_obs_error */
a_vis = _IQtoQ15(a);
/* a in formato Q.15 per condizioni del DAC */

int_plus_prop = a + K2_obs_error;
/* int_plus_prop = anello integrativo + anello proporzionale*/

KI_iq29 = _IQ(KI);
I_Ucfft_iq29 = _IQ(I_Ucfft);
/* KI e I_Ucfft sono definite nel DSP281x_Constant.h */

torque_feedforward = _IQmpyIQX(KI_iq29,29,I_Ucfft_iq29,29);
/* Torque feedforward */

omega_dot_hat = int_plus_prop + torque_feedforward;
/* Velocità angolare ad anello aperto, senza il contributo di K1_obs_error */
omega_dot_hat1 = omega_dot_hat << 16;
omega_dot_hat_vis = _IQtoQ15(omega_dot_hat1);
/* omega_dot_hat in formato Q.15 per condizioni del DAC */

omega_hat = omega_hat + omega_dot_hat;
/* Velocità angolare stimata non-migliorata, seconda trasformata z  $\{(z^{-1})/(1-Z^{-1})\}$  */
omega_hat1 = omega_hat << 16;
omega_hat_vis = _IQtoQ15(omega_hat1);
/* omega_hat in formato Q.15 per condizioni del DAC */

theta_hall = theta_hall + omega_hat_enh;
/* Inizio della terza trasformata z  $\{(1+Z^{-1})/(1-Z^{-1})\}$  */

omega_hat_enh = omega_hat + K1_obs_error;
/* Velocità angolare stimata migliorata */
omega_hat_enh1 = omega_hat_enh << 16;
omega_hat_enh_vis = _IQtoQ15(omega_hat_enh1);
/* omega_hat_enh in formato Q.15 per condizioni del DAC */

theta_hall = theta_hall + omega_hat_enh;
/* Posizione angolare stimata (continuazione della terza trasformata z  $\{(1+Z^{-1})/(1-Z^{-1})\}$  */

if (theta_hall >= PI_iq29)
    theta_hall = menoPI_iq29;
/* Ciclo creato per impostare la posizione angolare stimata tra -pigreco e +pigreco */

k_1_iq29 = _IQ(k_1);
/* k_1 è definita nel DSP281x_Constant.h */
K1_obs_error = _IQmpyIQX(k_1_iq29,29,obs_error_1,29);
/* K1_obs_error = k_1_iq29 * obs_error_1 */

theta_hall_pu = _IQdiv(theta_hall,PI_iq29);
/* Posizione angolare stimata in per unità, cioè tra -1 e +1 */

```

```

    theta_hall_int16 = _IQtoQ15(theta_hall_pu);
/* Si riporta in formato Q.15(int16) il valore di theta_hall_shiftato per poi essere messo come
argomento del seno e coseno del prodotto vettoriale; questo valore "theta_hall_int16" lo
possiamo considerare come la posizione angolare stimata ottenuta dall'osservatore ad
inseguimento vettoriale (VTO) */

    error_theta = theta_hall_pu - posiz_prov;
/* Errore di posizione angolare stimata, ottenuta sottraendo la posizione angolare stimata e la
posizione angolare di riferimento */
    error_theta_m = _IQmpyIQX(error_theta,29,_IQ(10),29);
/* Errore di posizione angolare stimata moltiplicata per 10, per poterla visualizzare
sull'oscilloscopio*/
    error_theta_vis = error_theta_m >> 16;
/* error_theta in formato Q.15 per condizioni del DAC */

GpioDataRegs.GPBCLEAR.bit.GPIOB3 = 1;
/* Porta digitale 4, impostata in basso sempre per calcolare il tempo di esecuzione di un ciclo
dell'osservatore */

}

i1 = i1 + 1;
if(i1==6)
    i1 = 1;

GpioDataRegs.GPBCLEAR.bit.GPIOB2 = 1;
/* Porta digitale 3, impostata in basso sempre per calcolare il tempo di esecuzione
dell'osservatore includendo posizione angolare di riferimento, Halpha e Hbeta e l'osservatore
*/

//-----
/*##### DIGITAL-ANALOG CONVERTER (DAC) #####*/
//-----

GpioDataRegs.GPBSET.bit.GPIOB4 = 1;
/* Porta digitale 5, impostata in alto per misurare il tempo di esecuzione del ADC */

DAC_buffer[0] = posiz_prov_vis;    // data to be displayed on DAC channel 1

DAC_buffer[1] = theta_hall_int16;  // data to be displayed on DAC channel 2

DAC_buffer[2] = Halpha;            // data to be displayed on DAC channel 3

DAC_buffer[3] = Hbeta;             // data to be displayed on DAC channel 4

DAC_buffer[4] = sin_theta_hall_vis; // data to be displayed on DAC channel 5

DAC_buffer[5] = cos_theta_hall_vis; // data to be displayed on DAC channel 6

DAC_buffer[6] = error_alpha_vis;   // data to be displayed on DAC channel 7

DAC_buffer[7] = error_beta_vis;    // data to be displayed on DAC channel 8

```

```

    GpioDataRegs.GPFSET.bit.GPIOF6 = 1;    // set CANTX pin
    AD7568_write(DAC_buffer);
    GpioDataRegs.GPFCLEAR.bit.GPIOF6 = 1;  // clear CANTX pin

    GpioDataRegs.GPBCLEAR.bit.GPIOB4 = 1;
    /* Porta digitale 5, impostata in basso sempre per calcolare il tempo di esecuzione del DAC */

    GpioDataRegs.GPBCLEAR.bit.GPIOB5 = 1;
    /* Porta digitale 6, impostata in basso sempre per calcolare il tempo che si mette la routine
    per effettuare le operazione di PWM, ABC -> ALPHA,BETA -> DQ, VTO e DAC */

    return;
}

#####
//                                     END OF PROGRAM
#####

```

Bibliografia

- **[Azi¹]** Università degli Studi di Udine, Facoltà di Ingegneria, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, “Dispensa sulle Macchine Sincrone a Magnete Permanente”, Udine, Italia, disponibile on-line sul sito "[http://www.diegm.uniud.it/petrella/Azionamenti%20Elettrici%20I/Complementary%20stuff%20chapter%205%20\(d\)%20-%20Azionamenti%20con%20MSMP.pdf](http://www.diegm.uniud.it/petrella/Azionamenti%20Elettrici%20I/Complementary%20stuff%20chapter%205%20(d)%20-%20Azionamenti%20con%20MSMP.pdf)"
- **[Azi²]** Politecnico di Milano, Facoltà di Ingegneria, Dipartimento di Elettrotecnica, “Dispensa sugli Azionamenti Brushless”, Milano, MI, Italia, disponibile on-line sul sito "<http://www.etec.polimi.it/IND32/It/AzionamentiElettrici/Materiale/AzionBrush.pdf>"
- **[Ana¹]** Data Sheet AD7568, “LC²MOS Octal 12-Bit DAC”, Analog Device Inc., disponibile on-line sul sito "<http://datasheet.digchip.com/041/041-01985-0-AD7568.pdf>".
- **[Dat¹]** “TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812. Digital Signal Processors Data Manual”, Texas Instruments Inc., Literature Number: SPRS174O, Aprile 2001 – Aggiornato Luglio 2007, disponibile on-line sul sito "<http://focus.ti.com/lit/ds/symlink/tms320f2812.pdf>"
- **[Deg¹]** M.W. Degner, “Flux, Velocity and Position Estimation in AC Machines Using Carrier Signal Injection”, Ph.D. Dissertation, University of Wisconsin-Madison, Madison, WI, USA, 1998.
- **[DeD¹]** Giulio De Donato, “Implementation end performance evaluation of a low cost surface-mounted pm machine drive using binary hall-effect sensors”, Ph.D. Dissertation, Sapienza – University of Rome, Rome, RM, ITALIA, 2006.
- **[Din¹]** Istituto Tecnico Industriale "Leonardo Da Vinci" Lanciano, “Controllo a Microprocessore della velocità di un motore in corrente continua a magnete permanente”, disponibile on-line sul sito "[http://utenti.quipo.it/itislanciano/Motorecc/ Link/Dinamo.htm](http://utenti.quipo.it/itislanciano/Motorecc/Link/Dinamo.htm)"
- **[Dis¹]** Politecnico di Milano, Facoltà di Ingegneria, Dipartimento di Elettrotecnica, “Dispensa sugli Azionamenti con Motori Brushless”, Milano, MI, Italia, disponibile on-line sul sito "<http://www.etec.polimi.it/IND32/It/AzionamentiVO/files/dispensa/Capitolo%206.doc>"
- **[Elt¹]** Eltra Silge Electronica S.A., “Dispensa sull'Encoder Incrementale”, Florida, BA, Argentina, disponibile sul sito "<http://www.silge.com.ar>"

- **[Ele¹]** E. Bassi, A. Bossi, “Elettrotecnica Pratica – Macchine ed azionamenti elettrici”, Volume terzo, Editoriale Delfino, 1990, ristampata 1995, Milano, MI, Italia.
- **[Fre¹]** Milan Brejl, Michal Princ, System Application Engineer, Roznov Czech System Center, “Using the PMSM Vector Control eTPU Function, Covers the MCF523x, MPC5500, and all eTPU-Equipped Devices”, Freescale Semiconductor - Application Note, Literature Number: AN2972, Rev.1 – 04/2006, disponibile on-line sul sito ["http://www.freescale.com/files/32bit/doc/app_note/AN2972.pdf"](http://www.freescale.com/files/32bit/doc/app_note/AN2972.pdf)
- **[Giu¹]** F. Giulii Capponi, G. De Donato, L. Del Ferraro, O. Honorati, M.C. Harke and R.D. Lorenz, “AC Brushless Drive With Low-Resolution Hall Effect Sensors for Surface-Mounted PM Machines”, in *IEEE Trans. Ind. Appl.*, Vol. 42, No. 2, pp 526-535, Mar./Apr. 2006.
- **[Hal¹]** “Sonde ad Effetto Hall”, disponibile on-line sul sito ["http://plent.altervista.org/automafile/sensore_ad_effetto_hall.htm"](http://plent.altervista.org/automafile/sensore_ad_effetto_hall.htm)
- **[Hal²]** Università degli Studi Mediterranea di Reggio Calabria, Facoltà di Ingegneria, DIMET, “Dispensa sulle Sonde ad effetto hall”, Reggio Calabria, Italia, disponibile on-line sul sito ["http://www.ing.unirc.it/portale/didattica/files_docenti/007300002020061016180100.pdf"](http://www.ing.unirc.it/portale/didattica/files_docenti/007300002020061016180100.pdf)
- **[Har¹]** M.C. Harke, G. De Donato, F. Giulii Capponi, T.R. Tesch and R.D. Lorenz “Implementation Issues and Performance Evaluation of Surface-Mounted PM Machine Drives with Hall-Effect Position Sensors and a Vector Tracking Observer”, in *Proc. IEEE IAS Annu. Meeting*, Tampa, FL, USA, Oct. 2006, cd-rom.
- **[Har²]** M.C. Harke, “Fundamental Sensing Issues in Motor Control”, Ph.D. Dissertation, University of Wisconsin-Madison, Madison, WI, USA, 2006.
- **[IQm¹]** “Virtual Floating Point Engine (IQmath Library) – Module user’s guide – C28x™ Foundation Software”, Texas Instruments Inc., Literature Number: SPRC087, Giugno 2002, Version 1.4.1., disponibile on-line sul sito ["http://focus.ti.com/docs/toolsw/folders/print/sprc087.html"](http://focus.ti.com/docs/toolsw/folders/print/sprc087.html)
- **[Jan¹]** P.L. Jansen and R.D. Lorenz, “Transducerless Field Orientation Concepts Employing Saturation Induced Saliencies in Induction Machines”, *IEEE Trans. Ind. Appl.*, Vol. 32, No. 6, pp 1380-1393, Nov./Dic. 1996.
- **[Ken¹]** R.M. Kennel, “Encoders for Simultaneous Sensing of Position and Speed in Electrical Drives with Digital Control” in *Proc. IEEE IAS Annu. Meeting*, Hong Kong, China, Oct. 2005, CD-ROM.

- **[Kim¹]** H. Kim, M.C. Harke and R.D. Lorenz, "Sensorless Control of Interior Permanent-Magnet Machine Drives with Zero-Phase Lag Position Estimation", *IEEE Trans. Ind. Appl.*, Vol. 39, No. 6, pp 1726-1733, Nov./Dic. 2003.
- **[Lor¹]** R.D. Lorenz, "Dynamics of Controlled Systems", Course Notes for ME 746, University of Wisconsin-Madison, Madison, WI, USA, 2002.
- **[Oga¹]** Katsuhiko Ogata, "Discrete Time Control Systems", 2nd edition, Prentice Hall Inc.; "Sistemas de Control en Tiempo Discreto", 2^{da} edicion, tradotto allo spagnolo per J.G. Aranda Perez, F. Rodriguez Ramirez e G. Sanchez Garcia, Prentice Hall Hispanoamerica S.A., Messico, 1996.
- **[QMA¹]** "Fixed Point Math Library (Qmath) – Module user's guide – C28xTM Foundation Software", Texas Instruments Inc., Literature Number: SPRC085, Maggio 2002, disponibile on-line sul sito "<http://focus.ti.com/docs/toolsw/folders/print/sprc085.html>"
- **[SiS¹]** S. Stocco, "Esercizi IG_meccatronica_e_automazione", Università degli Studi di Padova, Dipartimento di Tecnica e Gestione dei Sistemi Industriali, disponibile on-line sul sito "http://www.gest.unipd.it/esercizi/IG_meccatronica_e_automazione/Tesi_Simone_Stocco.doc"
- **[TeR¹]** "eZdspTM F2812 - Technical Reference", Spectrum Digital Inc., Stafford, TX, USA, 2003, disponibile on-line in "http://c2000.spectrumdigital.com/ezf2812/docs/ezf2812_techref.pdf"
- **[Tes¹]** T.R. Tesch, "Dynamic Torque Estimation in a Sensor Limited Environment", Ph.D. Dissertation, University of Wisconsin-Madison, Madison, WI, USA, 2005.
- **[TIN¹]** "TMS320x281x DSP Event Manager (EV) Reference Guide", Texas Instruments Inc., Literature Number: SPRU065D, Novembre 2004 – Aggiornato Agosto 2006, disponibile on-line sul sito "http://focus.ti.com/general/docs/techdocs_abstract.tsp?abstractName=spru065e"
- **[TIN²]** "TMS320x281x DSP Analog-to-Digital Converter (ADC) Reference Guide", Texas Instruments Inc., Literature Number: SPRU060D, Giugno 2002 – Aggiornato Luglio 2005, disponibile on-line sul sito "http://focus.ti.com/general/docs/techdocs_abstract.tsp?abstractName=spru060d"
- **[TIN³]** "TMS320x28xx, 28xxx DSP Serial Peripheral Interface (SPI) Reference Guide", Texas Instruments Inc., Literature Number: SPRU059D, Giugno 2002 – Aggiornato Novembre 2006, disponibile on-line sul sito "http://focus.ti.com/general/docs/techdocs_abstract.tsp?abstractName=spru059d"

BIBLIOGRAFIA

- **[Tra¹]** Prof. Paolo Mattavelli, Università degli Studi di Udine, Facoltà di Ingegneria, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, "Cenni su sensori e trasduttori", Udine, Italia, disponibile sul sito "http://www.diegm.uniud.it/mattavelli/elettronica_industriale/Dispense%20aggiuntive/Cenni%20su%20sensori%20e%20trasduttori.pdf"