

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Meccanica

Tesi di Laurea Magistrale

**SVILUPPO DI UNA DINAMO
FRENO PER UN BANCO
PROVA TRASMISSIONI
MECCANICHE**



Relatore

Prof. Andrea Mura

Candidato

Giada Colandrea

Dicembre 2020

Sommario

Il ruolo dell'automazione nel mondo di oggi ha un'influenza sempre più crescente sul modo di progettare, sviluppare e creare.

Semplificare il processo di scrittura e di sviluppo è il principale scopo con cui gli ingegneri si stanno confrontando.

A tale fine nel 2005, alcuni membri dell'Interaction Design Institute di Ivrea hanno ideato e sviluppato uno strumento per la prototipazione rapida, sia per scopi hobbistici che didattici e professionali.

Questo strumento, in linguaggio semplice, si chiama Arduino. Si tratta di una piattaforma hardware, composta da una serie di schede elettroniche dotate di un microcontrollore, abbinata ad un semplice ambiente di sviluppo integrato per la programmazione. Tutto il software a corredo è libero e gratuito, anche gli schemi circuitali sono distribuiti come hardware libero. Per questo motivo il suo utilizzo sta aumentando esponenzialmente.

Questo progetto propone la realizzazione di un'interfaccia di comando con cui gestire un sistema di controllo automatico PID, per poter variare due diversi parametri di controllo e valutarne i risultati.

Indice

Elenco delle figure	IV
1 Introduzione	1
1.1 Scopo	1
1.2 Materiali	1
2 Modello del Banco prova	5
2.1 Modello del Motore	6
2.2 Modello del Carico	8
2.3 Modello dell'accoppiamento	9
3 Tipi di controllo	11
3.1 Tecnica Di Controllo PWM	11
3.2 Tecnica Di Controllo PID	13
3.2.1 PID function su Arduino	13
4 Controllo del Motore	15
4.1 Controllo del motore ad anello aperto	15
4.1.1 Circuito di controllo del motore con configurazione ad anello aperto	17
4.1.2 Codice Arduino per il controllo del motore con configurazione ad anello aperto	18
4.2 Controllo del motore ad anello chiuso	21
4.2.1 Funzionamento di un Encoder	21
4.2.2 Funzionamento di una Dinamo Tachimetrica	27
4.2.3 Controllo PID di velocità con Dinamo Tachimetrica	31
5 Controllo della dinamo freno	37
5.1 Circuito usato per il controllo della coppia	39
5.2 Controllo del carico ad anello aperto	42
5.3 Controllo del carico ad anello chiuso	45

6	Risultati sperimentali del controllo di velocità	49
6.1	Grafici del controllo di velocità ad anello aperto	49
6.2	Grafici del controllo PID di velocità (closed-loop)	52
6.3	Grafici con effetto del carico sul controllo PID di velocità (closed-loop)	54
6.4	Confronto dei grafici di controllo PID di velocità (closed-loop) in presenza e in assenza di carico	56
6.4.1	Confronto tra le tensioni nel motore e nel carico	56
6.4.2	Confronto tra le correnti nel motore e nel carico	56
6.5	Controllo PID di velocità con profilo a step	59
6.6	Controllo PID di velocità con profilo sinusoidale	62
6.7	Controllo PID di velocità con profilo a trapezio	65
7	Risultati sperimentali del controllo di coppia	69
7.1	Grafici del controllo di coppia ad anello aperto	69
7.2	Grafici del controllo PID di coppia (close-loop)	72
7.2.1	Grafici del controllo PID di coppia variabile (close-loop) con tensione di alimentazione fissa	74
7.2.2	Grafici del controllo PID di coppia fissa (close-loop) con ten- sione di alimentazione variabile	77
7.3	Controllo PID di coppia con profilo a step	80
7.4	Controllo PID di coppia con profilo sinusoidale	83
7.5	Controllo PID di coppia con profilo a trapezio	86
8	Conclusioni	89
	Bibliografia	91

Elenco delle figure

1.1	Schema del banco prova	2
1.2	Motore con Dinamo tachimetrica usato nel Banco prova a bassa potenza	2
1.3	Scheda di Arduino Mega	3
1.4	Banco prova realizzato	4
2.1	Circuito finale con motore DC e generatore accoppiati	5
2.2	Circuito equivalente del motore DC	6
2.3	Caratteristica del motore	7
3.1	Caratteristiche PWM	11
3.2	Diversi dutycycle PWM	12
3.3	Schema a blocchi del PID	14
4.1	PWM nel motore	16
4.2	Schema a blocchi di controllo ad anello aperto	16
4.3	Primo approccio per il controllo del motore	17
4.4	Circuito ad anello aperto del motore con transistor e diodo	18
4.5	Schema del circuito di controllo del motore ad anello aperto	19
4.6	Comando analogico della tensione	19
4.7	Variazione automatica della tensione	20
4.8	Schema a blocchi del controllo ad anello chiuso	21
4.9	Schema di funzionamento di un Encoder	21
4.10	Encoder Assoluto	22
4.11	Encoder Incrementale	23
4.12	Segnali ad onde quadre din un Encoder Incrementale	24
4.13	Codice per comandare un Encoder (parte 1)	25
4.14	Codice per comandare un Encoder (parte 2)	26
4.15	Funzionamento di una dinamo tachimetrica	27
4.16	Schema di lettura e controllo PID di una dinamo tachimetrica	28
4.17	Codice di controllo ad anello aperto di una Dinamo Tachimetrica	30
4.18	Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 1)	32

4.19	Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 2)	33
4.20	Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 3)	34
4.21	schema di controllo PID di velocità con carico	35
5.1	Simbolo MOSFET IRL530N	39
5.2	Modulo MOSFET IRF520	40
5.3	Sensore di corrente ACS712 (5 AMP)	40
5.4	Circuito di controllo della coppia	41
5.5	Codice di controllo del carico ad anello aperto (prima parte)	42
5.6	Codice di controllo del carico ad anello aperto (seconda parte)	43
5.7	Codice di controllo del carico ad anello aperto (terza parte)	44
5.8	Codice di controllo del carico ad anello chiuso (prima parte)	45
5.9	Codice di controllo del carico ad anello chiuso (seconda parte)	46
5.10	Codice di controllo del carico ad anello chiuso (terza parte)	47
6.1	Velocità in funzione della tensione a cui viene alimentato il motore	50
6.2	Tensione effettiva letta nel motore a diverse tensioni di alimentazione	51
6.3	Corrente letta nel motore a diverse tensioni di alimentazione	51
6.4	Diversi controlli PID di velocità	52
6.5	Tensioni lette in diversi controlli PID di velocità	53
6.6	Correnti lette in diversi controlli PID di velocità	53
6.7	Diversi controlli PID di velocità con sistema collegato al carico massimo	54
6.8	Tensioni lette in diversi controlli PID di velocità con sistema collegato al carico massimo	55
6.9	Correnti lette in diversi controlli PID di velocità con sistema collegato al carico massimo	55
6.10	Confronto delle tensioni lette nel motore e nel carico in diversi controlli PID di velocità con sistema collegato al carico massimo	57
6.11	Confronto delle tensioni lette nel motore e nel carico in diversi controlli PID di velocità con sistema in assenza di carico	57
6.12	Confronto delle correnti lette nel motore e nel carico in diversi controlli PID di velocità con sistema collegato al carico massimo	58
6.13	Confronto delle correnti lette nel motore e nel carico in diversi controlli PID di velocità con sistema in assenza di carico	58
6.14	Controllo PID di velocità con profilo a step	59
6.15	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo a step	60
6.16	Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo a step	61
6.17	Coppia in un controllo PID di velocità con profilo a step	61

6.18	Controllo PID di velocità con profilo sinusoidale	62
6.19	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo sinusoidale	63
6.20	Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo sinusoidale	63
6.21	Coppia in un controllo PID di velocità con profilo sinusoidale	64
6.22	Controllo PID di velocità con profilo a trapezio	65
6.23	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo a trapezio	66
6.24	Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo a trapezio	67
6.25	Coppia in un controllo PID di velocità con profilo a trapezio	67
7.1	Massima coppia in funzione della tensione a cui viene alimentato il motore	70
7.2	Variazione della velocità al variare del PWM che comanda la coppia	71
7.3	Coppia esistente (in assenza di carico) in funzione della tensione a cui viene alimentato il motore	71
7.4	Controllo PID di coppia di 30 Nmm in funzione di diverse tensioni di alimentazione	72
7.5	Diversi controlli PID di coppia	73
7.6	Velocità in diversi controlli PID di coppia alla tensione di alimentazione di 17 V	73
7.7	Tensioni lette nel Motore variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)	74
7.8	Correnti lette nel Motore variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)	75
7.9	Tensioni lette nel Carico variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)	75
7.10	Correnti lette nel Carico variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)	76
7.11	Tensioni lette nel Motore variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm	77
7.12	Correnti lette nel Motore variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm	78
7.13	Tensioni lette nel Carico variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm	78
7.14	Correnti lette nel Carico variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm	79
7.15	Controllo PID di coppia con profilo a step	80
7.16	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a step	81

7.17	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a step	81
7.18	Velocità in un controllo PID di coppia con profilo a step	82
7.19	Controllo PID di coppia con profilo sinusoidale	83
7.20	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo sinusoidale	84
7.21	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo sinusoidale	84
7.22	Velocità in un controllo PID di coppia con profilo sinusoidale	85
7.23	Controllo PID di coppia con profilo a trapezio	86
7.24	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a trapezio	87
7.25	Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a trapezio	87
7.26	Velocità in un controllo PID di coppia con profilo a trapezio	88

Capitolo 1

Introduzione

1.1 Scopo

Questa tesi ha il fine di progettare e realizzare un banco prova per alberi scanalati tramite l'uso del software Arduino. In particolare viene spiegato come il banco prova deve essere composto e il codice implementato per l'azionamento. Infine vengono riportati i risultati delle prove sperimentali effettuate. Il banco sarà progettato in bassa potenza e poi successivamente scalato su una potenza maggiore.

Lo scopo preciso di questa tesi è di sviluppare due controlli: il primo sulla velocità di rotazione del sistema e il secondo sulla coppia resistente applicata.

1.2 Materiali

Lo schema del banco prova è riportato in fig. 1.1 ed è composto dai seguenti componenti:

- *motore DC*: per azionare il banco; siccome in ingresso è collegato all'alimentazione, riceve una tensione (energia elettrica) e la trasforma in velocità di rotazione (energia meccanica). Motore usato nel banco prova in fig. 1.2.
- *Encoder o Dinamo Tachimetrica*: come sensore di velocità per controllo ad anello chiuso sul motore
- *Dinamo freno*: dispositivo uguale al motore DC usato come generatore, ovvero con entrata e uscita invertite (riceve una certa velocità e trasmette un voltaggio, quindi trasforma energia meccanica in energia elettrica). Viene usato per creare una coppia resistente nel sistema.
- *Arduino*: piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore, che permette il controllo e il pilotaggio del banco prova



Figura 1.1: Schema del banco prova

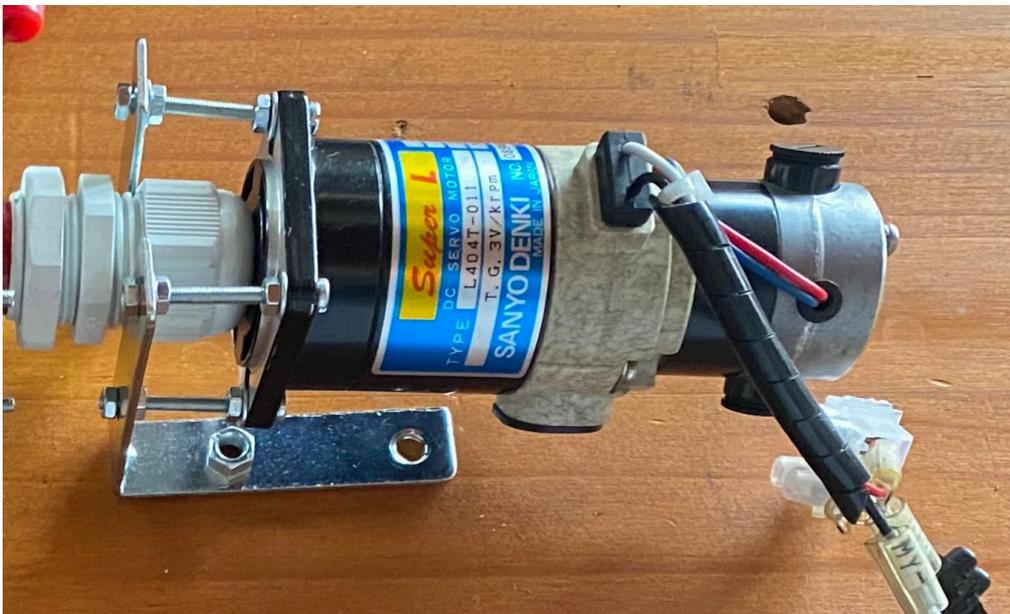


Figura 1.2: Motore con Dinamo tachimetrica usato nel Banco prova a bassa potenza

- *Sensore di corrente:* per la misura della corrente. Utile nel caso in cui il calcolo indiretto non dia risultati precisi.



Figura 1.3: Scheda di Arduino Mega

- *Elettronica di potenza*: circuito che può anche essere composto solo da un tasto o da un transistor, permette la variazione di tensione nel circuito con diverse finalità
- *Resistenza*: rappresenta il carico applicato alla dinamo freno grazie al quale è possibile variare la coppia: una resistenza nulla rappresenta una coppia massima ed è quindi indispensabile nel sistema
- *Alimentazione elettrica*: per la scheda Arduino e per l'elettronica di potenza

In particolare, il motore con dinamo tachimetrica in dotazione, in fig. 1.2 è il modello L404T-011 SANYO DENKI, con le seguenti caratteristiche:

- Potenza: 40 W
- Corrente: 2.7 A
- Tensione: 24 V
- k_e : 3 V/Krpm

Il banco prova realizzato in bassa potenza è riportato in fig. 1.4.

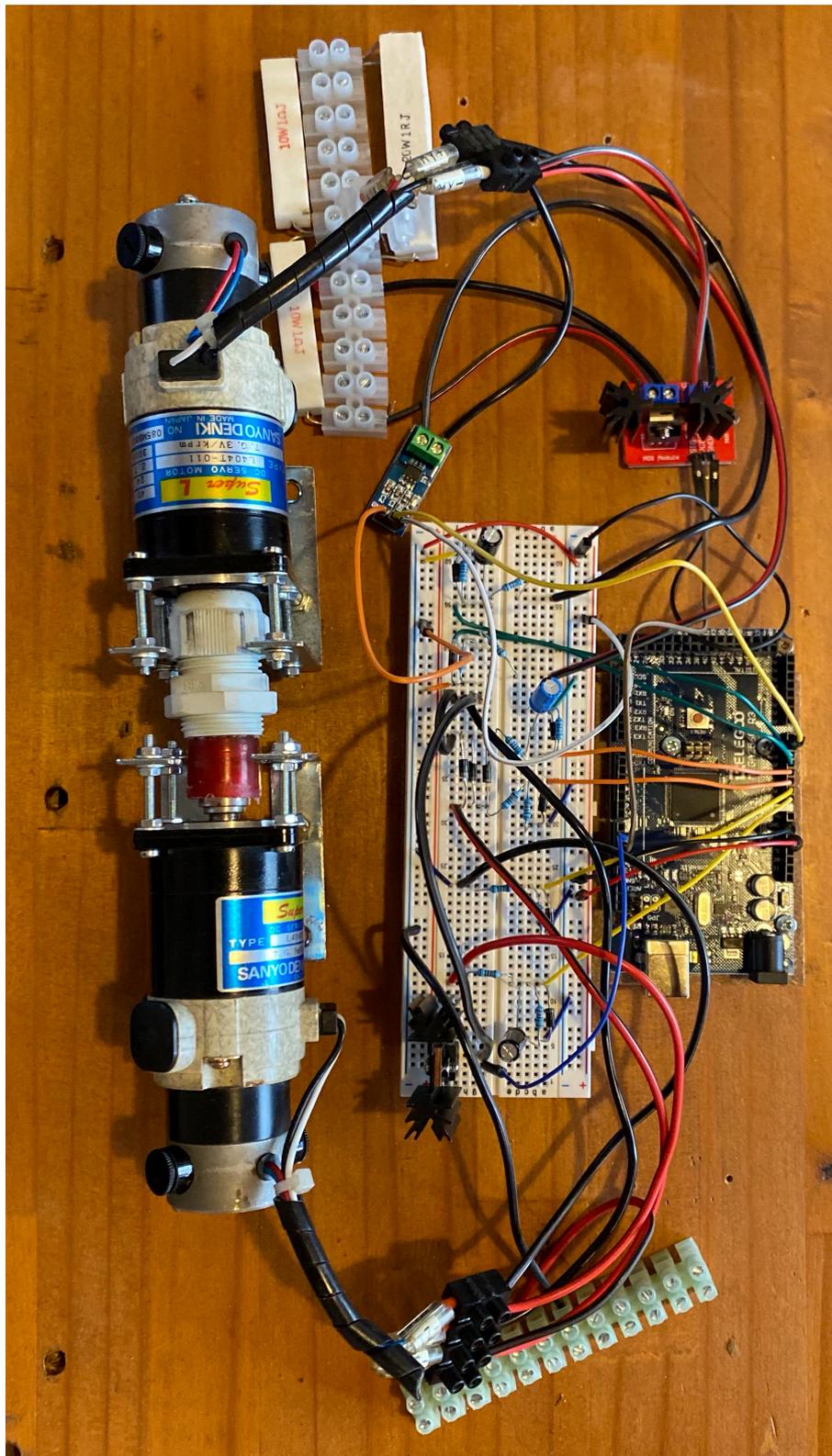


Figura 1.4: Banco prova realizzato

Capitolo 2

Modello del Banco prova

In questo capitolo vengono formulati i modelli matematici che caratterizzano i componenti principali del banco prova:

- motore DC: si ha in ingresso una tensione (energia elettrica) e in uscita una velocità di rotazione (energia meccanica)
- carico: è un motore DC usato come generatore, quindi si ha in ingresso una velocità di rotazione (energia meccanica) e in uscita una tensione (energia elettrica)
- accoppiamento: motore e carico vengono uniti e viene eseguita la modellizzazione finale

Si riporta lo schema del circuito finale in fig. 2.1.

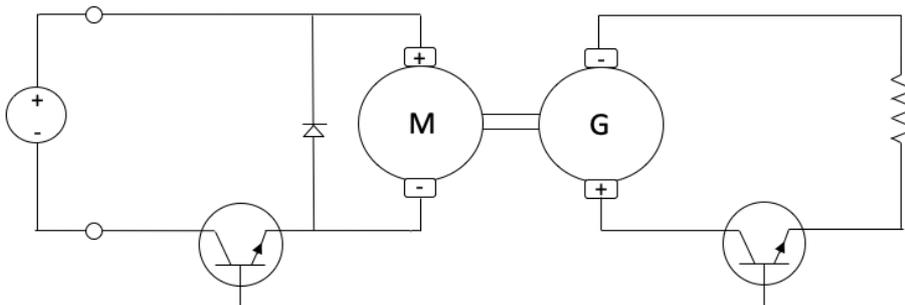


Figura 2.1: Circuito finale con motore DC e generatore accoppiati

Nei seguenti modelli non viene considerato l'effetto dei transistor in quanto si intende prima formulare un modello base; i transistor verranno implementati successivamente per il controllo di velocità e coppia.

2.1 Modello del Motore

Il motore usato in questo progetto è un motore in corrente continua che si può modellare tramite il seguente circuito equivalente in fig. 2.2.[1]

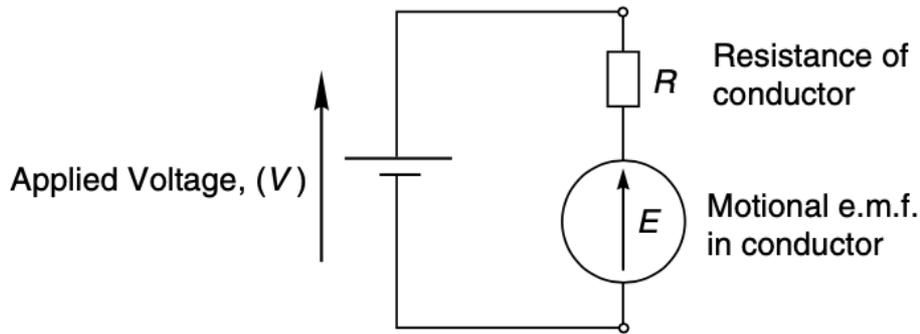


Figura 2.2: Circuito equivalente del motore DC

L'equazione caratteristica del motore, in condizioni stazionarie, è:

$$V - E = R_i I \quad (2.1)$$

in cui

- V è la tensione applicata al motore
- E è la forza elettromotrice indotta
- R_i è la resistenza interna del motore
- I è la corrente che circola nel motore

Nel motore elettrico a corrente continua sono anche valide le seguenti equazioni:

$$E = k_e \omega \quad (2.2)$$

$$T = k_T I \quad (2.3)$$

con

$$k_T (Nm/A) = k_e (V/rad/s)$$

Dalle equazioni sopra si può ricavare la formulazione della velocità del motore:

$$V - k_e \omega = R_i \frac{T}{k_T}$$

e quindi:

$$\omega = \frac{1}{k_e} \left(V - R_i \frac{T}{k_T} \right) \quad (2.4)$$

oppure la formulazione della coppia:

$$T = \frac{k_T}{R_i} (V - k_e \omega) \quad (2.5)$$

L'equazione 2.4 è graficamente rappresentata in fig. 2.3 e servirà per l'impostazione del controllo di velocità del motore. [1]

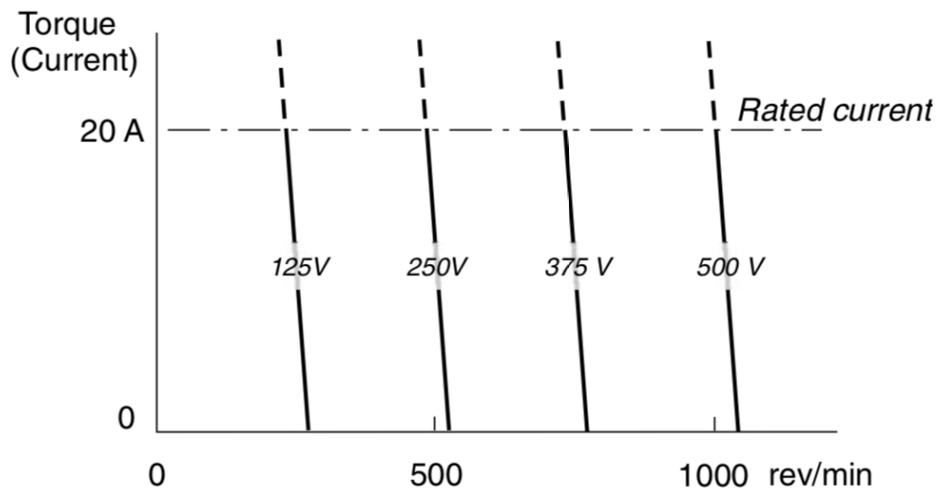


Figura 2.3: Caratteristica del motore

2.2 Modello del Carico

Il carico è un motore DC usato in maniera inversa. Se infatti nel motore si ha una trasformazione dell'energia da forma elettrica (tensione e corrente) a forma meccanica (coppia e velocità), nel carico al contrario si ha una trasformazione da energia meccanica a energia elettrica. Il carico è pertanto un motore utilizzato come generatore DC, che è a sua volta collegato a una resistenza elettrica R_e utilizzata per dissipare l'energia elettrica generata.

L'equazione caratteristica del carico quindi è

$$E - V = R_i I \quad (2.6)$$

in cui:

- E è la forza elettromotrice indotta (positiva per convenzione)
- V è la tensione in uscita dal carico
- R_i è la resistenza interna del motore
- I è la corrente che circola nel carico

Poiché il generatore è collegato a una resistenza esterna, è valida anche la seguente equazione:

$$V = R_e I \quad (2.7)$$

Considerando inoltre le relazioni tra forza elettromotrice indotta e velocità e tra coppia e corrente (già espresse nel modello elettrico del motore e qui nuovamente valide), si ricavano le seguenti equazioni caratteristiche del carico:

$$I = \frac{k_e \omega}{R_e + R_i} \quad (2.8)$$

$$T = \frac{k_e k_T \omega}{R_e + R_i} \quad (2.9)$$

Le equazioni di cui sopra daranno la base per l'impostazione del controllo di coppia che verrà eseguito sul sistema.

2.3 Modello dell'accoppiamento

Per la caratterizzazione fisica del banco prova si devono unire le due equazioni 2.4 e 2.9 considerando che la velocità e la coppia del motore siano uguali a quelle del carico (in condizioni stazionarie e trascurando gli attriti dell'accoppiamento). Quindi si ottiene:

$$\omega = \frac{R_e + R_i}{R_e + 2R_i} \frac{V}{k_e} \quad (2.10)$$

$$T = \frac{k_T}{R_e + 2R_i} V \quad (2.11)$$

Inoltre si può trovare la formulazione della potenza:

$$P = T\omega = \frac{(R_e + R_i)}{(R_e + 2R_i)^2} V^2 \quad (2.12)$$

Capitolo 3

Tipi di controllo

3.1 Tecnica Di Controllo PWM

Il segnale PWM (pulse width modulation, ovvero modulazione a larghezza di impulso) è un segnale digitale in cui il periodo dell'impulso alla massima ampiezza varia rispetto il periodo del segnale. Questo tipo di segnale permette di ottenere un segnale analogico variando il periodo dell'impulso.

Il segnale PWM ha due caratteristiche (riportate in fig. 3.1), l'ampiezza massima e il duty cycle, la prima dipende dalla grandezza del segnale mentre il secondo parametro descrive la durata dell'impulso al massimo livello.

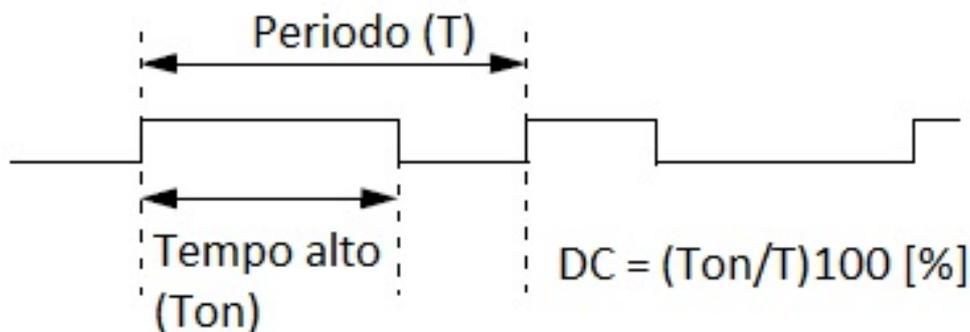


Figura 3.1: Caratteristiche PWM

Il duty cycle è dato dal periodo del segnale a massimo livello diviso il periodo del segnale moltiplicato per 100, quindi è un valore che si misura in percentuale. Si può anche esprimere come numero decimale senza moltiplicare per 100. I comandi relativi in Arduino sono evidenti nella seguente immagine. Come si può notare dall'immagine, se il dutycycle è pari allo 0% si avrà un valore nullo del segnale

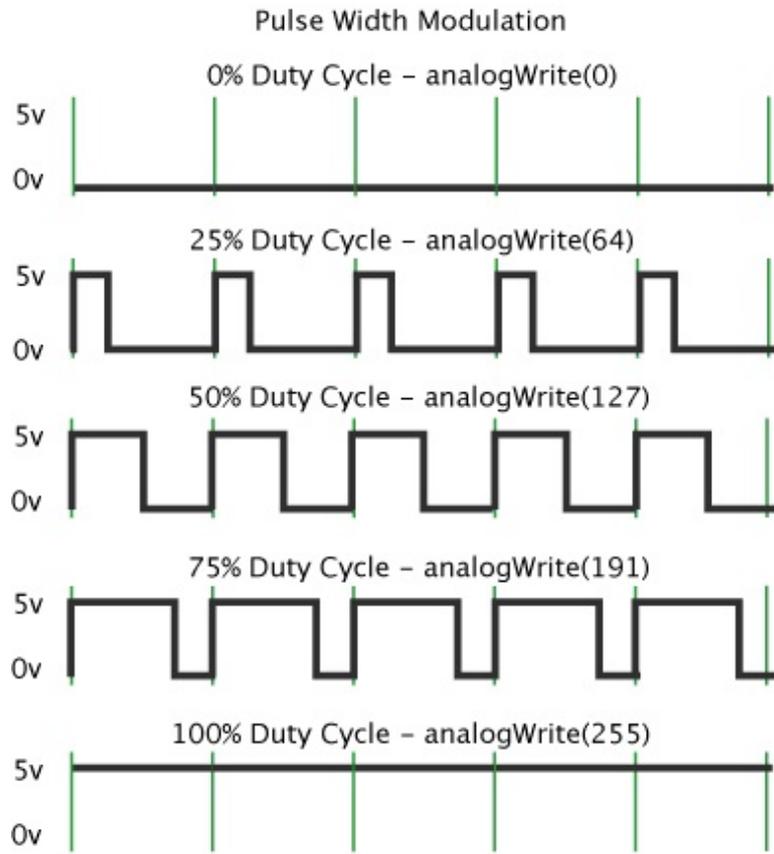


Figura 3.2: Diversi dutycycle PWM

analogico; aumentando il dutycycle aumenta il valore del segnale fino ad arrivare ad un dutycycle del 100 % che permette al segnale di avere il valore massimo.

3.2 Tecnica Di Controllo PID

Il PID è un sistema in “retroazione” negativa ampiamente impiegato nei sistemi di controllo, di gran lunga più comune nell’industria. PID è un acronimo dove P significa “Proporzionale”, I indica “Integrale” e infine D per “Derivativo”, queste sono le 3 parti necessarie a ottenere un controllo con retroazione, il quale riduce automaticamente l’errore tra il valore di uscita atteso rispetto a quello effettivamente ottenuto. Quindi esso regola l’uscita in base alle seguenti azioni:

- *Azione proporzionale*: Valore del segnale di errore

$$A_p = K_p * e(t)$$

è pari a una costante per l’errore

- *Azione integrale*: Valori passati del segnale di errore

$$A_i = K_i * \int e(t) dt$$

è pari a una costante per l’integrale dell’errore rispetto al tempo

- *Azione derivativa*: Quanto velocemente il segnale di errore varia

$$A_d = K_d \frac{de(t)}{dt}$$

è pari a una costante per la derivata dell’errore rispetto al tempo

Queste azioni, calcolate singolarmente, vengono poi sommate algebricamente nella seguente formula:

$$output = A_p + A_i + A_d$$

3.2.1 PID function su Arduino

Questa funzione crea un controllo PID collegato ai specifici Input, Output e al valore di Regolazione.

La sintassi é PID(&Input, &Output, &Setpoint, Kp, Ki, Kd, Direction) in cui:

- *Input*: le variabili che si cercano di controllare (double)
- *Output*: le variabili che vengono aggiustate dal pid (double)
- *Setpoint*: le variabili che si vogliono mantenere costanti con il controllo (double ≥ 0)

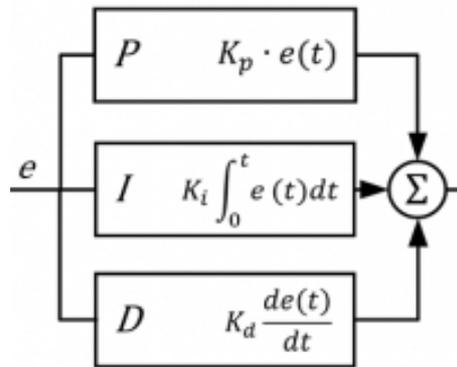


Figura 3.3: Schema a blocchi del PID

- K_p, K_i, K_d : parametri relativi alla regolazione PID (Tuning Parameters) ($double \geq 0$)
- *Direction*: sia DIRECT o REVERSE. Determina in quali direzioni deve muoversi l'output quando si incontra un errore. DIRECT è il più comune

Capitolo 4

Controllo del Motore

Il primo passo da compiere per lo sviluppo del banco prova è l'osservazione del comportamento del motore, lo sviluppo di un circuito che permetta il collegamento del motore all'alimentazione (configurazione ad anello aperto) e successivamente lo sviluppo di un controllo del motore tramite encoder o dinamo tachimetrica (configurazione ad anello chiuso). La caratteristica del motore, riportata in fig. 2.3, mostra la relazione data dall'eq. (2.4) ovvero (in assenza di coppia)

$$\omega = \frac{V}{k}$$

Secondo tale relazione, modificando la tensione all'ingresso del motore è possibile ottenere una variazione direttamente proporzionale della velocità di rotazione dell'albero in uscita al motore stesso.

Il metodo di controllo PWM può essere applicato ai motori DC perché sono dei dispositivi induttivi: l'induzione è un processo per cui una corrente elettrica variabile che scorre in un filo è in grado di generare un campo magnetico variabile. Quando si dà elettricità ad un motore, una bobina di filo di rame avvolta strettamente all'interno del motore crea un campo magnetico. Questo campo fa girare l'albero del motore. Nel nostro caso, il potere induttivo del motore permette al dispositivo di avere un periodo di tempo, successivamente all'apertura del circuito, in cui la corrente continua a scorrere. Ciò permette, in tempi molto brevi, di sottoporre il dispositivo ad alternanza di presenza e assenza di tensione e di avere come risultato una corrente media più o meno costante, esattamente come si può vedere in fig. 4.1.

4.1 Controllo del motore ad anello aperto

Una configurazione ad anello aperto è un sistema con nessun collegamento di retroazione dove si ha in ingresso dell'energia elettrica, un azionamento che mette in moto

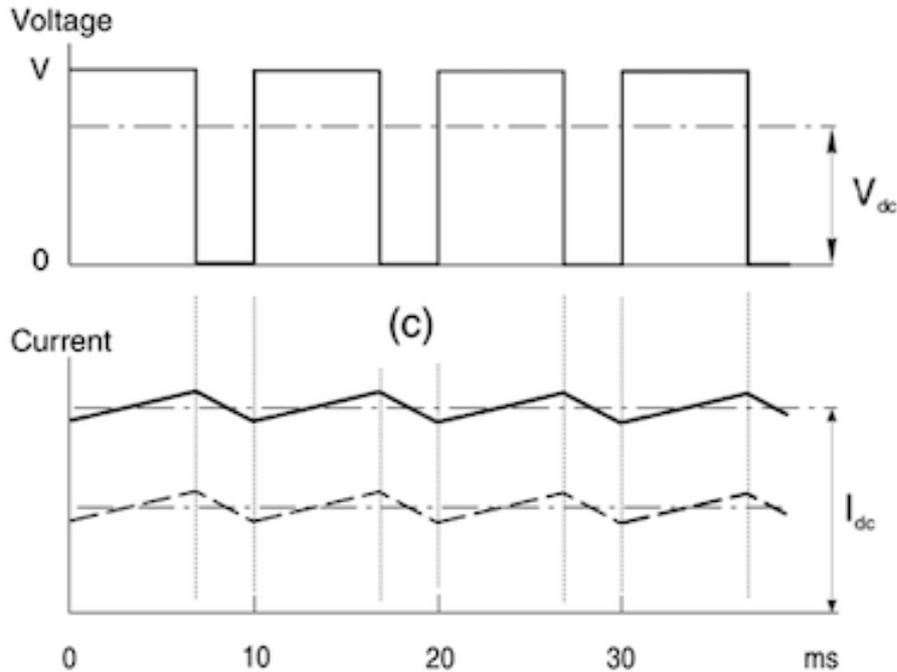


Figura 4.1: PWM nel motore

il motore tramite una differenza di potenziale e in uscita dell'energia meccanica caratterizzata dalla velocità di rotazione. Lo schema è riportato in fig. 4.2:

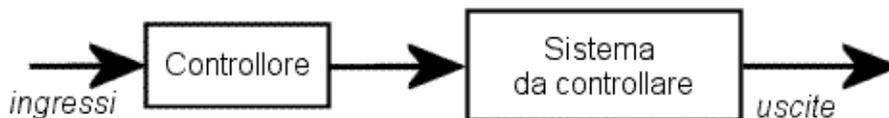


Figura 4.2: Schema a blocchi di controllo ad anello aperto

Si potrebbe pensare, come si può vedere in fig. 4.3, di mettere un tasto all'ingresso dell'alimentazione del motore, pilotarlo con Arduino affinché cambiando la tensione con controllo PWM si ottenga una variazione di velocità del motore.

Ciò non può essere fatto perché la scheda di Arduino non può reggere tutta la potenza necessaria, ma può fornire solo 40 mA. Ciò che può essere utile è quindi un transistor.

I transistor sono componenti che permettono di controllare alti carichi di corrente e di tensione tramite la bassa corrente di uscita di un piedino di Arduino.

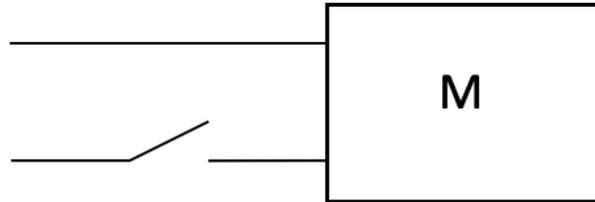


Figura 4.3: Primo approccio per il controllo del motore

Esistono molti tipi diversi, tutti basati sullo stesso principio. Si può pensare ad un transistor come ad un piedino digitale. Quando si fornisce tensione a uno dei piedini del transistor, chiamato gate (porta), si chiude il circuito tra gli altri due piedini, chiamati source (sorgente) e drain (scarico); in tal modo con Arduino si può accendere e spegnere un motore di elevata tensione o corrente.

Il circuito che permette il controllo ad anello aperto è quello riportato in fig. 4.4 in cui a sinistra si può vedere il caso in cui il transistor è attivato e la corrente delle armature fluisce nel generatore, mentre a destra il transistor è aperto e la corrente delle armature fluisce nel diodo; questo quindi permette al motore di continuare a ruotare per un breve tempo.

I diodi sono dei dispositivi che permettono alla corrente elettrica di scorrere solo in una direzione. Riescono a fare questo perché sono polarizzati: significa che è importante la direzione con cui sono inseriti nel circuito; collocati in un modo, permettono il passaggio di corrente attraverso di essi mentre, se inseriti nella direzione opposta, la bloccano.

4.1.1 Circuito di controllo del motore con configurazione ad anello aperto

Il circuito, il cui schema viene riportato in fig. 4.5 è stato realizzato con i seguenti elementi:

- *transistor di tipo MOSFET*
- *Diodo IN4007*
- *Motore da 6/9 V*
- *Connettore per batteria*

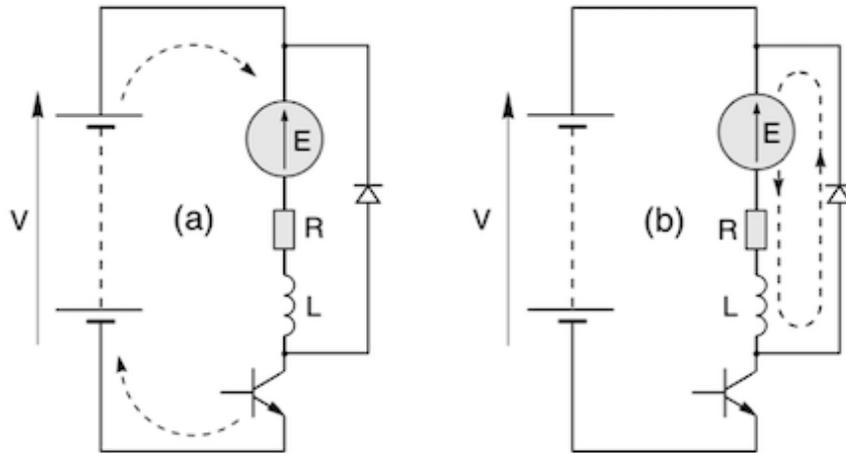


Figura 4.4: Circuito ad anello aperto del motore con transistor e diodo

- Batteria da 9 V
- Ponticelli

4.1.2 Codice Arduino per il controllo del motore con configurazione ad anello aperto

Per il controllo del sistema sono stati utilizzati due tipi di codice:

- A) comando analogico della tensione

In questo caso il controllo è stato implementato con il comando `digitalWrite(pin, value)`. La sintassi del comando prevede che al posto di "pin" si metta il numero del pin a cui è collegato il transistor (nel nostro caso il pin 9). Al posto di "value" ci sono due possibili soluzioni:

- si possono inserire i valori "HIGH" e "LOW", che significano rispettivamente acceso alla tensione massima o spento
- si possono inserire dei valori compresi tra 0 e 255 che ci restituiscono dei valori intermedi di velocità

Come si può vedere in fig. 4.6, è stata usata la seconda opzione. In particolare sono stati usati i valori che si possono trovare in fig. 4.1.

- B) variazione automatica della tensione

In questo caso si è proceduto scrivendo un codice che cambiasse automaticamente la velocità del motore. Ciò è stato possibile aumentando la tensione

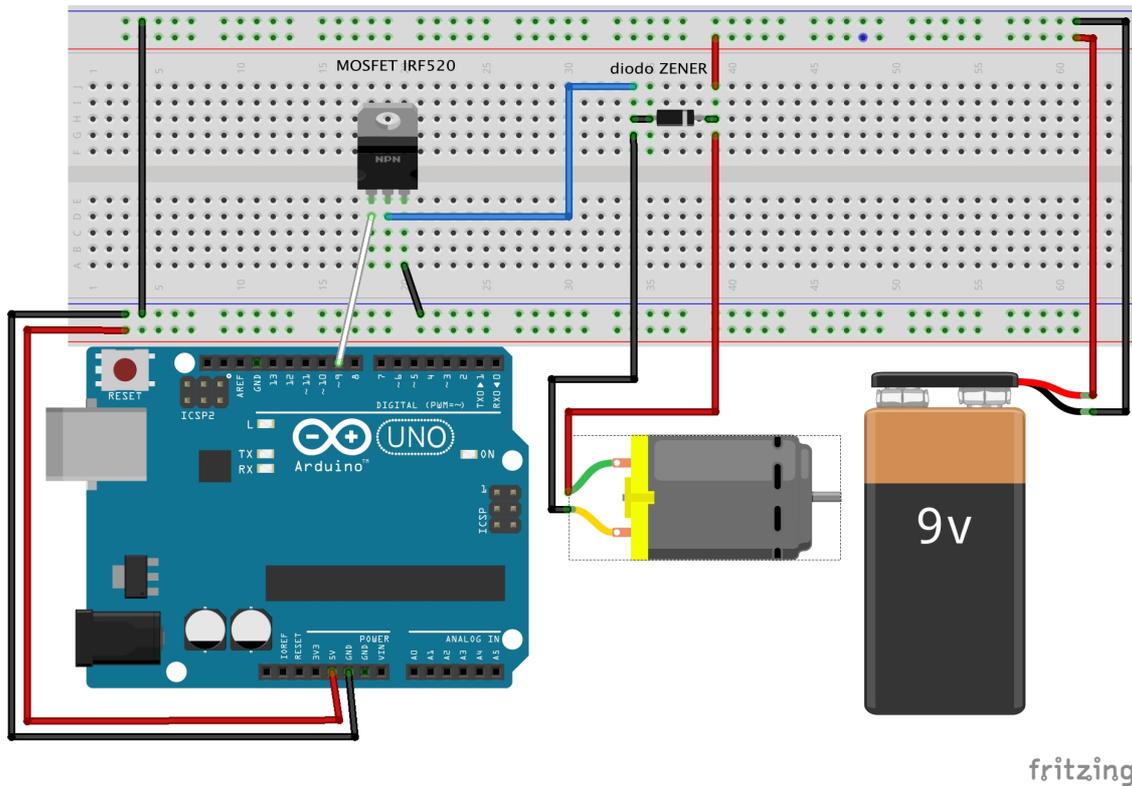


Figura 4.5: Schema del circuito di controllo del motore ad anello aperto

```

controllo_motore_anello_aperto §
const int motorePin = 9;

void setup() {
  pinMode(motorePin,OUTPUT);
}

void loop() {
  digitalWrite(motorePin,10);
}

```

Figura 4.6: Comando analogico della tensione

fornita di un valore fisso ad ogni loop del codice. Quindi, una volta che il programma è stato caricato sulla scheda di Arduino, si è potuto osservare il motore partire a velocità nulla, aumentare la velocità fino ad un massimo per poi diminuire la velocità fino a fermarsi nuovamente, per ricominciare a girare. Il codice è riportato in fig. 4.7.

```
controllo_motore_anello_aperto_2
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  delay(100);
}
```

Figura 4.7: Variazione automatica della tensione

4.2 Controllo del motore ad anello chiuso

Il controllo ad anello chiuso è stato effettuato con due strumenti differenti in modo da poter supportare un banco prova sia a basse che ad alte potenze:

- Dinamo tachimetrica per banchi prova a bassa potenza
- Encoder per banchi prova a potenza più alta

Lo schema a blocchi del controllo ad anello chiuso è riportato in fig. 4.8.

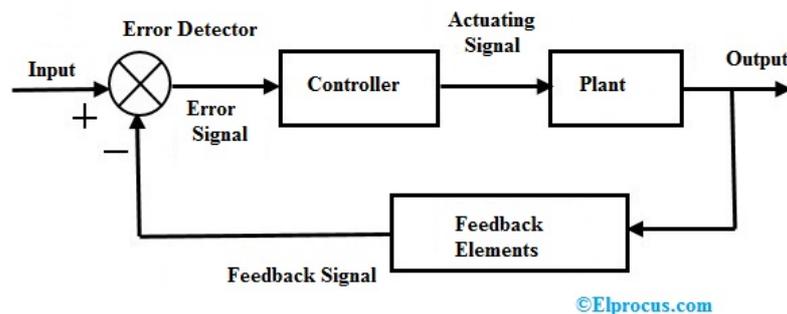


Figura 4.8: Schema a blocchi del controllo ad anello chiuso

4.2.1 Funzionamento di un Encoder

In questa sezione è stato testato il corretto funzionamento dell'encoder tramite un codice che permettesse di visualizzare la ricezione e il comportamento del singolo dispositivo separato dall'intero sistema.

L'encoder è un apparato elettromeccanico che converte la posizione angolare del suo asse rotante in un segnale elettrico digitale. Esso è un trasduttore di posizione e

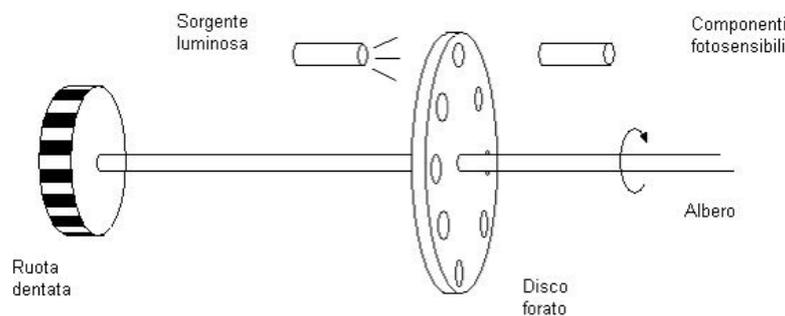


Figura 4.9: Schema di funzionamento di un Encoder

di velocità il cui schema di funzionamento è riportato in fig. 4.9 : l'albero trasmette

il moto al disco coronato da settori circolari opachi e trasparenti; a lato del disco è situata una sorgente luminosa e nel lato opposto i componenti fotosensibili. Con la rotazione del disco si hanno delle ripetute interruzioni del fascio di luce che li percorre. Questa variazione di luminosità genera l'uscita di un segnale digitale a seconda della posizione dell'albero. Ciò permette di avere oltre ad un segnale di posizione anche un segnale di velocità. Esistono due tipi di Encoder:

- *Encoder Assoluto* : Trasduttore di posizione costituito da un elemento mobile a forma di disco, su cui sono state ricavate per fotoincisione maschere di zone opache e trasparenti. Questo scorre davanti a testine di lettura costituite da fotoemettitori e fotorivelatori, che generano segnali dipendenti dallo spostamento. Negli encoder assoluti non si ha una semplice successione di zone chiare e di zone scure, ma le zone chiare e quelle scure, disposte su più piste concentriche, rappresentano, lette su un raggio, una parola in codice binario. Il codice binario non è tuttavia adatto ad essere utilizzato in un encoder. Infatti succede spesso che passando da un numero binario al successivo, si presenta una variazione di più di una cifra. Quindi può accadere che durante un passaggio da un settore all'altro, si possa avere una successione casuale delle cifre con una conseguente lettura di valori non corrispondente al vero. Per ovviare a questo problema si preferisce utilizzare il codice grey, nel quale

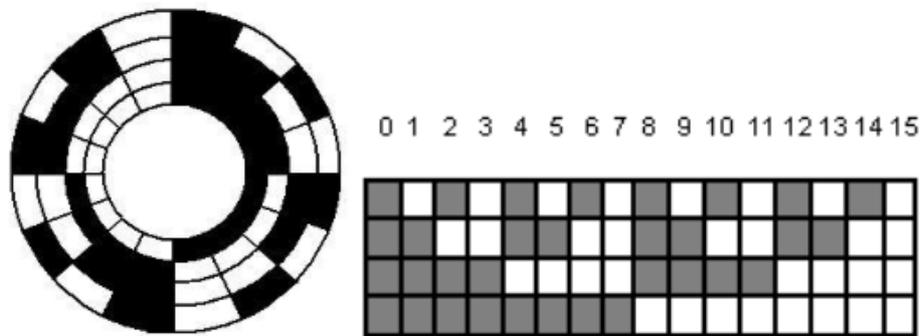


Figura 4.10: Encoder Assoluto

il passaggio da un numero al successivo avviene sempre variando un'unica cifra binaria. Per migliorare la risoluzione di un encoder ottico assoluto, oltre ad aumentare il numero delle piste, si può accoppiare l'encoder all'organo da controllare tramite un demoltiplicatore in modo che, ad esempio, ad un giro della vite, corrispondano tre giri del disco decodificatore. Il demoltiplicatore impiegato deve essere di altissima precisione ed avere la ripresa automatica dei giochi. La figura sottostante rappresenta un disco mobile, di un encoder assoluto, che utilizza un codice binario a quattro bit.

- *Encoder Incrementale* :L'encoder di tipo incrementale è il più semplice, in quanto fornisce solamente informazioni di tipo on-off per tenere sotto controllo parametri quali velocità, direzione o distanza. Esso è un trasduttore di posizione, il cui sistema di intercettazione è costituito da un disco sul quale sono state ricavate due corone concentriche di finestrelle rettangolari. Tali finestrelle, poste una sopra l'altro sono sfasate fra di loro di un quarto di passo. Ai lati del sistema di intercettazione sono presenti, da una parte due fotoemettitori e dall'altra due fotorivelatori (un fotoemettitore + un fotorivelatore per ciascuna corona concentrica di finestrelle). I fotoemettitori generano un treno di impulsi il cui numero è pari al numero delle zone trasparenti, alternate alle scure, intercettate dal blocco emettitore-ricevitore. Il conteggio di questi impulsi, amplificati dall'elettronica di controllo dell'encoder, consente di individuare la rotazione compiuta dal disco ed il corrispondente spostamento. Le due corone concentriche poste sul disco una sopra l'altra, permettono inoltre all'encoder di capire quando il disco gira in senso orario e quando antiorario. La risoluzione di un encoder incrementale è data dal rapporto tra una rotazione di 360° ed il numero degli impulsi : $R=360/4n$. In fig. 4.11, è rappresentato il disco mobile di un encoder incrementale. La risoluzione

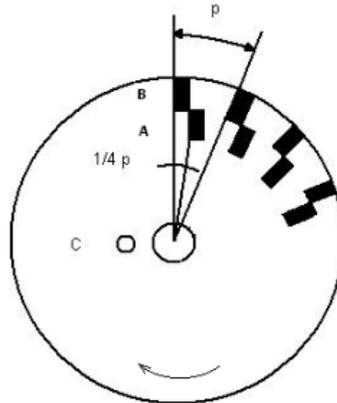


Figura 4.11: Encoder Incrementale

può però, essere migliorata se le tensioni di uscita dei fototransistor vengono opportunamente squadrate e vengono contati sia i fronti di salita, sia quelli in discesa. Dire che le tensioni di uscita devono essere squadrate vuol dire avere un circuito elettronico che rende costante la tensione di uscita per tutto l'intervallo di tempo in cui la luce dei fotoemettitori, colpiscono i fotorivelatori in modo da avere una forma d'onda come quella nella fig. 4.12.

L'encoder usato nel banco prova è un encoder di tipo incrementale rotativo MINIROD 421 di Heidenhain. Questo dispositivo richiede una potenza di

$$P = 5V \pm 5\%$$

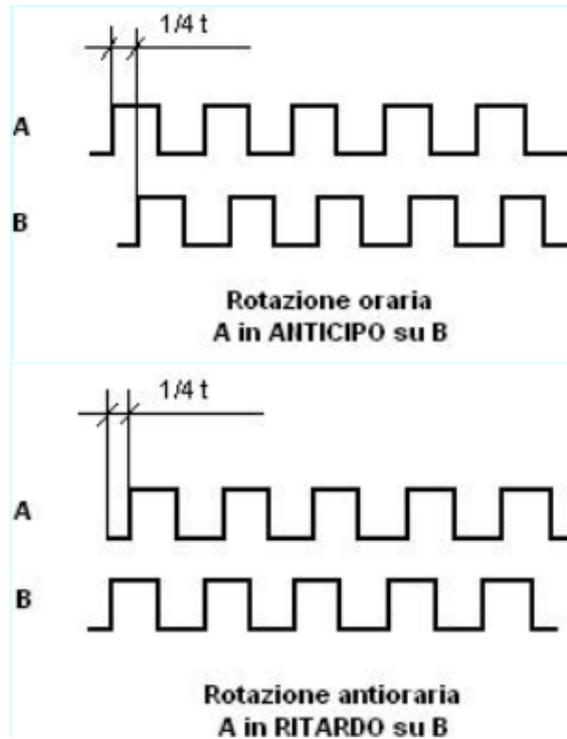


Figura 4.12: Segnali ad onde quadre di un Encoder Incrementale

e una corrente

$$I \geq 160mA$$

Arduino non può fornire una corrente maggiore di 40 mA quindi è stata utilizzata un'alimentazione esterna. Il collegamento è di tipo elettronico, si ha un giunto con 12 pin i quali sono da collegare nel seguente modo:

- *pin 2*: cavo colore blu - alimentazione 5 V
- *pin 3* : cavo colore rosso - pin 4 di Arduino
- *pin 5*: cavo colore marrone - pin 3 di Arduino
- *pin 8*: cavo colore grigio - pin 2 di arduino
- *pin 10*: cavo colore verde - 0 V (GND)
- *pin 11* : cavo colore bianco - 0 V (GND)
- *pin 12* : cavo colore verde - alimentazione 5 V

```

pid_motore_2 §
#include <EnableInterrupt.h>
#include <digitalWriteFast.h>

#define encoderA 2
#define encoderB 3
#define encoderZ 4

volatile int countA = 0;
volatile int countB = 0;
volatile int cumulativeCountA = 0;
volatile int cumulativeCountB = 0;
int pulsesPerRev = 14400;
int Dir = 0; // 1 = CW
            // 0 = Stationary
            // -1 = CCW

void setup() {
  Serial.begin(9600);
  pinMode(encoderA, INPUT);
  pinModeFast(encoderB, INPUT);
  pinMode(encoderZ, INPUT);
  enableInterrupt(encoderA, pulseA, RISING);
  enableInterrupt(encoderB, pulseB, RISING);
  enableInterrupt(encoderZ, pulseZ, RISING);
}

void loop() {
  //Serial.print(countA);
  Serial.print('\t');
  //Serial.print(countB);
  Serial.print('\t');
  //Serial.print(cumulativeCountA);
  Serial.print('\t');
  //Serial.print(cumulativeCountB);
  Serial.print('\t');
  Serial.println(Dir);
}

```

Figura 4.13: Codice per comandare un Encoder (parte 1)

Codice di controllo di un Encoder

Il codice implementato per il controllo è riportato in fig. 4.13 e in fig. 4.14

In questo codice sono state usate le seguenti librerie:

- *EnableInterrupt.h* : Con il termine interrupt (interruzione) intendiamo un

```
void checkDirection(){
  if((bool) digitalReadFast(encoderB) == HIGH){
    Dir = 1;
  }
  else{
    Dir = -1;
  }
}

void pulseA(){
  checkDirection();
  countA += Dir;
  cumulativeCountA += Dir;
}

void pulseB(){
  countB += Dir;
  cumulativeCountB += Dir;
}

void pulseZ(){
  countA = 0; //reset counters at "home" reference point
  countB = 0;
}
```

Figura 4.14: Codice per comandare un Encoder (parte 2)

segnale asincrono che indica la necessità di “attenzione” da parte di una periferica collegata ad Arduino. L’interrupt viene generato quando si verifica una variazione di stato su uno dei piedini di Arduino. Normalmente il microcontrollore esegue all’interno del `loop()` in modo sequenziale e ripetitivo le istruzioni in esso inserite, ma quando si verifica un interrupt viene interrotto il flusso delle istruzioni all’interno del `loop()` ed invocate altre routine (create dall’utente). Quando le routine terminano il flusso del programma prosegue normalmente. L’utilizzo dell’interrupt è particolarmente utile quando abbiamo la necessità di gestire in background alcune routine, ovvero eseguire istantaneamente un’operazione nel caso si manifesti un evento asincrono esterno. Capita spesso di avere la necessità di controllare lo stato di diversi pin di input, in questa situazione potrebbe capitare che il programma non si accorga del cambiamento di stato su uno dei pin, utilizzando gli interrupt si evita questo tipo di errore.

- *digitalWriteFast.h* : questa libreria invece permette una commutazione più veloce dei diversi pin.

4.2.2 Funzionamento di una Dinamo Tachimetrica

La dinamo tachimetrica è un sensore di velocità con un principio di funzionamento simile a quello del motore elettrico DC, che consiste nell'invertire ingresso con uscita: si considera in ingresso la velocità angolare e come uscita la tensione. Questo funzionamento è permesso dalla presenza di un circuito elettrico, solidale alla parte mobile, immerso in un campo magnetico fisso generato da un magnete permanente. Quando il circuito ruota, una tensione viene generata. Tale tensione in uscita è proporzionale alla velocità di rotazione in ingresso, come in fig. 4.15.

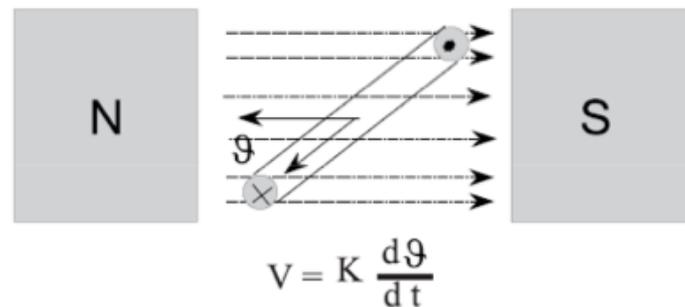


Figura 4.15: Funzionamento di una dinamo tachimetrica

La dinamo tachimetrica è un sensore assoluto, a basso costo, attivo (ovvero senza alimentazione) e adatto ad ambienti ostili.

Nonostante abbia anche diversi lati negativi rispetto ad un Encoder (che sarà descritto in seguito), verrà usata la dinamo tachimetrica nel primo step della progettazione del banco prova. Gli svantaggi sono:

- *affidabilità e qualità ridotta del segnale di uscita*: ciò è dovuto alla presenza di spazzole striscianti sul circuito elettrico; il segnale di uscita è caratterizzato quindi da oscillazioni generate durante il passaggio delle spazzole sulle zone isolate del collettore
- *Facilmente deteriorabile* a causa dei contatti striscianti
- *Ripple strutturale sull'uscita*

Elettronica di potenza per il controllo di una Dinamo Tachimetrica

L'elettronica di potenza è descritta in fig. 4.16.

Siccome il segnale della dinamo tachimetrica ricevuto su Arduino è maggiore di 5 V, è stato necessario creare un partitore di tensione: due resistenze da 10 kOhm e un diodo IN4007. Il segnale proveniente dalla dinamo tachimetrica, in questo modo, viene diviso a metà e può essere elaborato dalla scheda di Arduino. Per

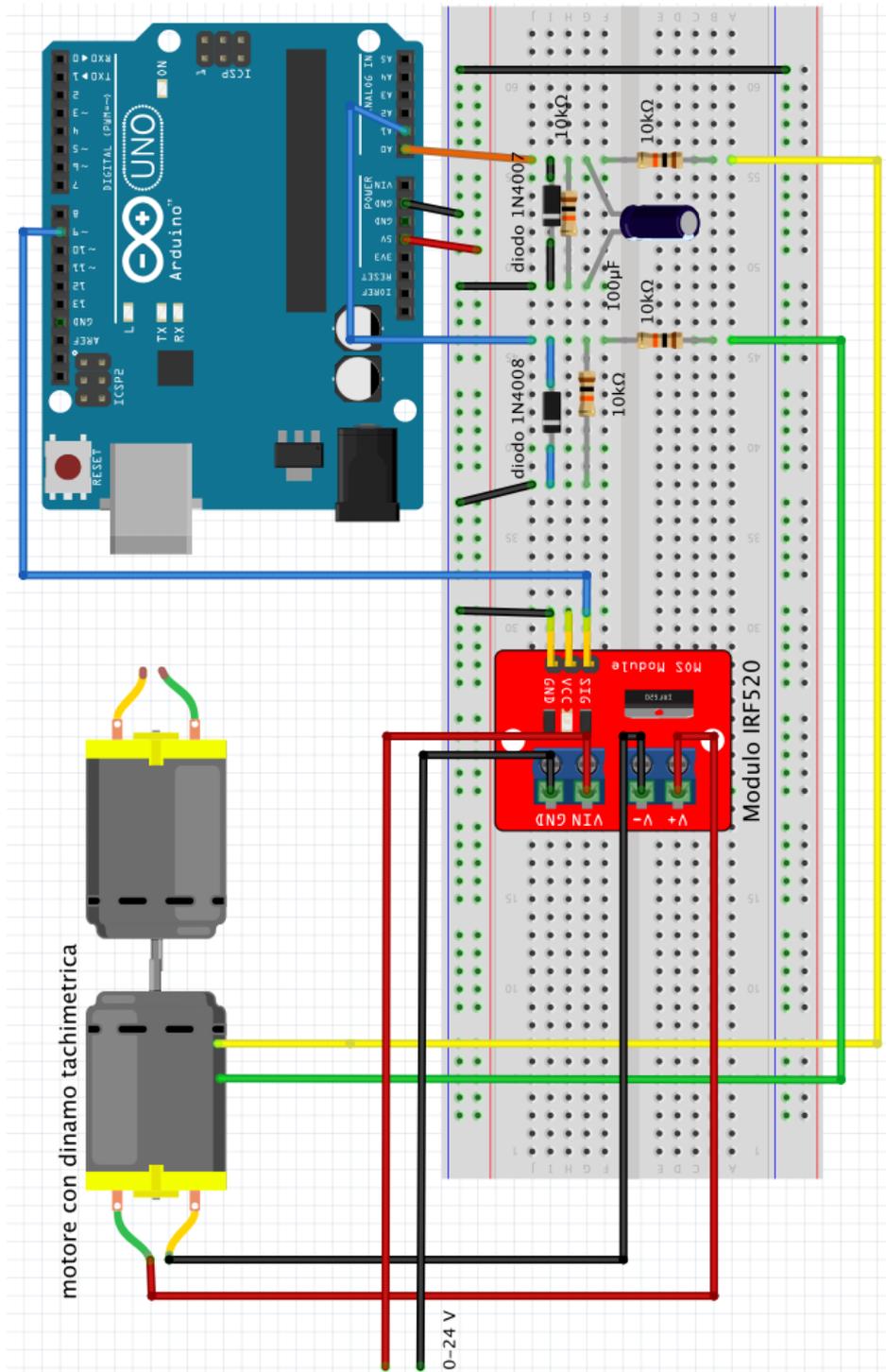


Figura 4.16: Schema di lettura e controllo PID di una dinamo tachimetrica

calcolare la velocità in funzione del voltaggio in uscita dalla dinamo è stata usata la seguente formula:

$$\omega = \frac{1}{Ke} * V$$

con

$$Ke = \frac{1000rpm}{3V}$$

Per il comando di tensione è stato scelto come transistor il Modulo IRF520. Per maggiori dettagli riguardo la scelta dei transistor si rimanda alla sezione 5.1.

Codice di controllo ad anello aperto di una Dinamo Tachimetrica

Il codice implementato è riportato in fig. 4.17.

L'input in ingresso è la tensione che si vuole applicare al sistema: siccome la tensione massima assorbita dal motore è 24 V, sarà possibile impostare delle tensioni inferiori a tale soglia. La tensione impostata viene quindi convertita in PWM. Il PWM letto dalla funzione AnalogWrite() è un valore compreso tra 0 e 255 quindi è necessario scalare la tensione con la seguente formula

$$PWM_V = \frac{V_{applicata}}{V_{massima}} * 255 \quad (4.1)$$

Il codice applica al motore l'alimentazione e legge dalla dinamo tachimetrica la velocità ottenuta. E' possibile quindi leggere diverse velocità impostando voltaggi differenti.

Il controllo della velocità ad anello aperto consiste nel comunicare al transistor collegato al motore un dutycycle proporzionale alla velocità voluta, secondo l'equazione:

$$PWM_\omega = \frac{\omega_{voluta}}{\omega_{massima}} * 255 \quad (4.2)$$

con

$$\omega_{massima} = \frac{V_{alimentatore}}{24[V]} * 3000[rpm]$$

Il controllo ad anello aperto, di cui l'equazione 4.2 non tiene conto dei disturbi (es: presenza di una coppia, di resistenze, attriti, cadute di tensione sull'alimentazione, ecc..).

I risultati sono riportati nei grafici della sezione 6.1.

```

#define dinamoPOS A0
#define dinamoNEG A1
#define motore 9
#define LetturaTensioneMotorePOS A2
#define LetturaTensioneMotoreNEG A3

//valori da cambiare
float tensioneApplicata = 0;
float tensioneTrasformatore = 17;
//
volatile int tensioneDinamoPOS = 0;
volatile int tensioneDinamoNEG = 0;
double velocita;
unsigned long t1;
int pwm = 0;
int TensioneMotoreLettaPOS, TensioneMotoreLettaNEG;
float resistenzaInternaMotore = 4.5; //ohm

void setup() {

  TCCR1B = TCCR1B & 0b11111000 | 0x3;

  Serial.begin(9600);
  pinMode(dinamoPOS, INPUT);
  pinMode(dinamoNEG, INPUT);
  pinMode(LetturaTensioneMotoreNEG, INPUT);
  pinMode(LetturaTensioneMotorePOS, INPUT);
  pinMode(motore, OUTPUT);
}

void loop() {
  //ALIMENTAZIONE
  TensioneMotoreLettaPOS = analogRead(LetturaTensioneMotorePOS);
  TensioneMotoreLettaNEG = analogRead(LetturaTensioneMotoreNEG);
  float TensioneMotoreLettaVolt = (TensioneMotoreLettaPOS - TensioneMotoreLettaNEG)*30/1023.0; //V
  //   Serial.print("Tensione_di_alimentazione_del_motore_voluta:\t");
  //   Serial.print(tensioneApplicata);
  //   Serial.print(" ");
  //   Serial.print("Tensione_di_alimentazione_del_motore:\t");
  //   Serial.println(TensioneMotoreLettaVolt);

  float CorrenteMotore = (TensioneMotoreLettaVolt-8*velocita/1000.0)/resistenzaInternaMotore; //A
  //   Serial.print("Corrente_di_alimentazione_del_motore:\t");
  //   Serial.println(CorrenteMotore);

  pwm = tensioneApplicata*255/tensioneTrasformatore; //numero tra 0 e 255
  analogWrite(motore, pwm);

  //DINAMO
  tensioneDinamoPOS = analogRead(dinamoPOS); //numero tra 0 e 1023
  tensioneDinamoNEG = analogRead(dinamoNEG);
  float tensioneDinamoVolt = (tensioneDinamoPOS - tensioneDinamoNEG)*(10/1023.0); // V
  velocita = (tensioneDinamoPOS - tensioneDinamoNEG)*(1000/3.0)*(10/1023.0); //rpm

  //Serial.print("pwm:\t");
  //Serial.println(pwm);

  //   Serial.print("velocità:\t");
  //   Serial.println(velocita); //rpm

  //Serial.print("tensione_della_dinamo_achimetrica:\t");
  //Serial.println(tensioneDinamoVolt);
}

```

Figura 4.17: Codice di controllo ad anello aperto di una Dinamo Tachimetrica

4.2.3 Controllo PID di velocità con Dinamo Tachimetrica

Il controllo PID applicato al banco prova dotato di Dinamo Tachimetrica é riportato in fig. 4.18, in fig. 4.19 e in fig. 4.20.

In ingresso ciò che si può impostare è la velocità che si vuole ottenere dal banco prova. Il codice preleva tale informazione, la confronta con la velocità registrata dalla dinamo tachimetrica e modifica il PWM applicato al motore, cercando di avvicinarsi, ad ogni loop, al valore della velocità ricercata. Lo schema dell'elettronica di potenza è riportato in fig. 4.16 e ovviamente è lo stesso usato nel controllo ad anello aperto.

Il controllo di velocità ad anello chiuso somma il valore ottenuto con il controllo ad anello aperto, calcolato con l'equazione 4.2, con il risultato ottenuto dal controllo PID:

$$PWM_{\omega} = PWM_{outputPID} + \frac{\omega_{voluta}}{\omega_{massima}} * 255 \quad (4.3)$$

con

$$\omega_{massima} = \frac{V_{alimentatore}}{24[V]} * 3000[rpm]$$

In questa formulazione la parte del PID permette di compensare i disturbi, ma è legata alla precisione della dinamo tachimetrica.

Il controllo PID viene tarato con il metodo Ziegler e Nichols, che prevede di:

- Chiudere il loop di controllo inserendo un regolatore puramente proporzionale con basso guadagno K_p
- Aumentare progressivamente K_p fino a che si innesca un'oscillazione permanente
- Iniziare ad aumentare il K_i fermandosi una volta che il massimo rendimento è stato ottenuto

Tarando in questo modo il controllo PID sono stati trovati sperimentalmente i valori ottimali :

- pidtime: 0.1 ms
- K_p : 0.01
- K_i : 0.2
- K_d : 0.001

```

#include <PID_v1.h>
#define dinamoPOS A0
#define dinamoNEG A1
#define LetturaTensioneMotorePOS A2
#define LetturaTensioneMotoreNEG A3
#define motore 9
#define carico 10
#define LetturaTensioneCaricoPOS A4
#define LetturaTensioneCaricoNEG A5
#define LetturaCorrenteCarico A6

//valori che si possono cambiare
double velocitaVoluta = 00; // [rpm] velocità da impostare
int pwmCarico = 0; //coppia da impostare
double Kp=0.01, Ki=0.2, Kd=0.001;
unsigned long pidtime = 0.1;
float tensioneTrasformatore = 17;
//
volatile int tensioneDinamoPOS = 0;
volatile int tensioneDinamoNEG = 0;
volatile int TensioneMotoreLettaPOS,TensioneMotoreLettaNEG;
volatile int TensioneCaricoLettaPOS,TensioneCaricoLettaNEG;
volatile int CorrenteCaricoLetta;
double velocitaOttenuta, tensioneMotore;
float tensioneMotoreVolt = 0;
unsigned long t1;
int pwmMotore = 0;
float Samples=0.0, CorrenteCaricoLettaMedia=0.0;
int x=0;
float CorrenteCarico=0;
float velmax = tensioneTrasformatore*3000/24.0;
float resistenzaInternaMotore = 5; //ohm

PID myPID(&velocitaOttenuta, &tensioneMotore, &velocitaVoluta, Kp, Ki, Kd, DIRECT);

PID myPID(&velocitaOttenuta, &tensioneMotore, &velocitaVoluta, Kp, Ki, Kd, DIRECT);

void setup() {

  TCCR1B = TCCR1B & 0b11111000 | 0x03;

  Serial.begin(9600);
  pinMode(dinamoPOS,INPUT);
  pinMode(dinamoNEG,INPUT);
  pinMode(LetturaTensioneMotorePOS,INPUT);
  pinMode(LetturaTensioneMotoreNEG,INPUT);
  pinMode(LetturaTensioneCaricoPOS,INPUT);
  pinMode(LetturaTensioneCaricoNEG,INPUT);
  pinMode(LetturaCorrenteCarico,INPUT);
  pinMode(motore,OUTPUT);
  pinMode(carico,OUTPUT);
  digitalWrite(motore, HIGH);

```

Figura 4.18: Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 1)

```

myPID.SetSampleTime(pidtime);
myPID.SetOutputLimits(0, 255);
myPID.SetMode(AUTOMATIC);
}

void loop() {

    analogWrite(carico, pwmCarico);
    analogWrite(motore, pwmMotore);

    //ALIMENTAZIONE
    TensioneMotoreLettaPOS = analogRead(LetturaTensioneMotorePOS);
    TensioneMotoreLettaNEG = analogRead(LetturaTensioneMotoreNEG);

    float TensioneMotoreLettaVolt = (TensioneMotoreLettaPOS - TensioneMotoreLettaNEG)*30/1023.0;
    // Serial.print("Tensione_di_alimentazione_del_motore:\t");
    // Serial.println(TensioneMotoreLettaVolt);
    float CorrenteMotore = (TensioneMotoreLettaVolt-8*velocitaOttenuta/1000.0)/resistenzaInternaMotore;
    // Serial.print("Corrente_di_alimentazione_del_motore:\t");
    // Serial.println(CorrenteMotore);
    //CARICO
    TensioneCaricoLettaPOS = analogRead(LetturaTensioneCaricoPOS);
    TensioneCaricoLettaNEG = analogRead(LetturaTensioneCaricoNEG);

    float TensioneCaricoLettaVolt = (TensioneCaricoLettaPOS - TensioneCaricoLettaNEG)*30/1023.0;
    // Serial.print("Tensione_che_passa_nel_carico:\t");
    // Serial.println(TensioneCaricoLettaVolt);

    if (x<100) {
        CorrenteCaricoLetta = analogRead(LetturaCorrenteCarico);
        Samples = Samples + CorrenteCaricoLetta;
        x++;
    }
    else{
        x=0;
        CorrenteCaricoLettaMedia=Samples/100;
        CorrenteCarico = (2.5 - ((CorrenteCaricoLettaMedia*(5.0/1024.0)))/0.185;
    // Serial.print("Corrente_che_passa_nel_carico:\t");
    // Serial.println(CorrenteCarico);
        Samples=0;
    }

    if ((millis() - t1) >= pidtime) {

```

Figura 4.19: Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 2)

```

if ((millis() - t1) >= pidtime) {
    //DINAMO
    tensioneDinamoPOS = analogRead(dinamoPOS); //numero tra 0 e 1023
    tensioneDinamoNEG = analogRead(dinamoNEG);
    float tensioneDinamoVolt = (tensioneDinamoPOS - tensioneDinamoNEG)*(10/1023.0);
    //      Serial.print("tensione_della_dinamo_tachimetrica:\t");
    //      Serial.println(tensioneDinamoVolt);

    velocitaOttenuta = (tensioneDinamoPOS - tensioneDinamoNEG)*(1000/3.0)*(10/1023.0); //rpm
    //      Serial.print("Velocità_voluta:\t");
    //      Serial.print(velocitaVoluta); // rpm
    //
    //      Serial.print(" ");

    //      Serial.print("Velocità_ottenuta:\t");
    //      Serial.println(velocitaOttenuta); // rpm

    myPID.Compute();
    pwmMotore = tensioneMotore + velocitaVoluta*255.0/velmax;

    tensioneMotoreVolt = pwmMotore*tensioneTrasformatore/255; // V
    //      Serial.print("tensione_del_motore:");
    //      Serial.println(tensioneMotoreVolt);

    t1 = millis();
}
}

```

Figura 4.20: Codice di controllo velocità con dinamo tachimetrica ad anello chiuso (parte 3)

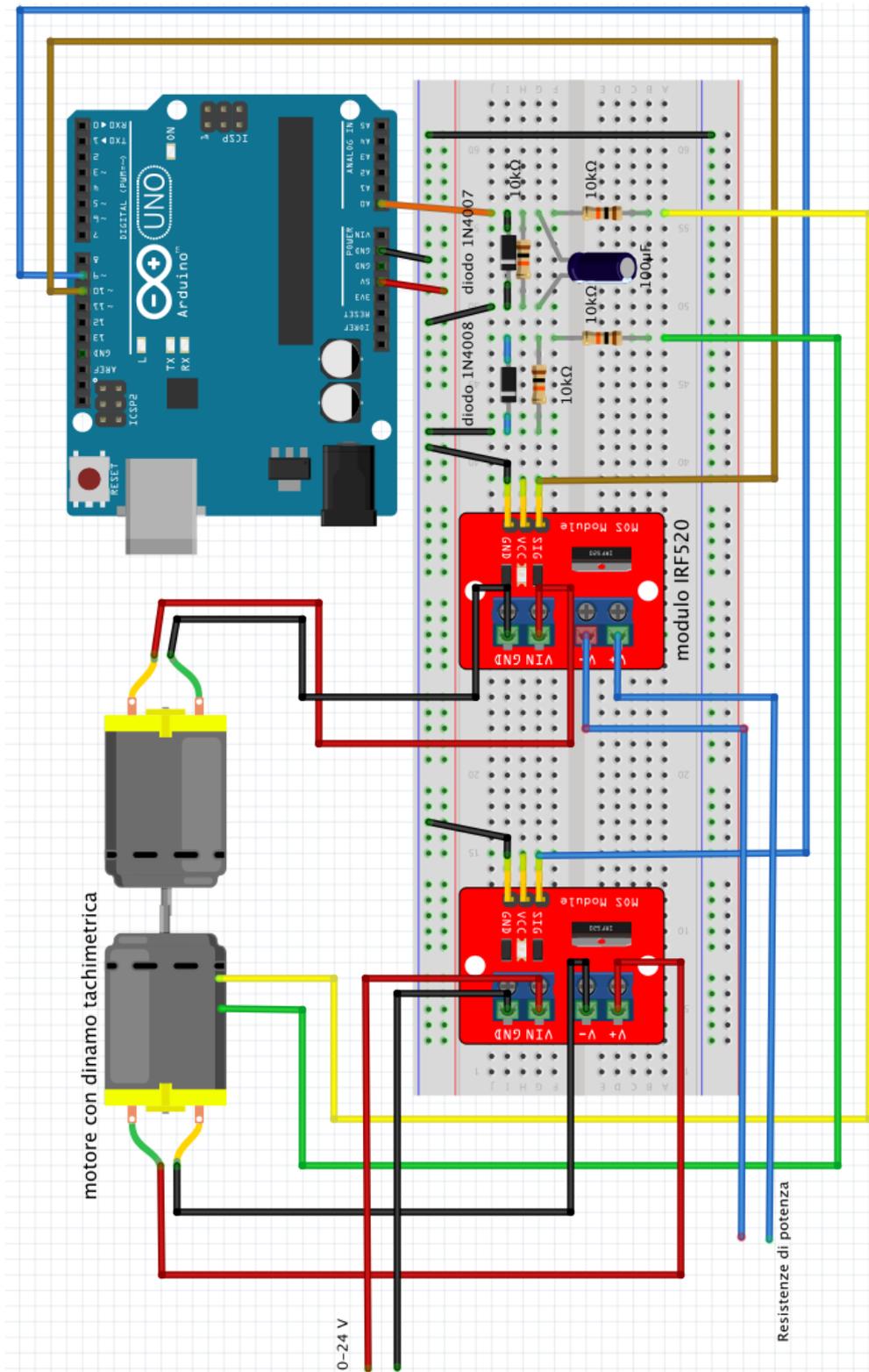


Figura 4.21: schema di controllo PID di velocità con carico

Capitolo 5

Controllo della dinamo freno

Il primo passo per lo sviluppo del controllo del carico è la realizzazione di un circuito che permetta di inserire un transistor all'uscita del generatore, collegato ad una resistenza esterna. Siccome l'uso del torsionometro era impossibile nella costruzione del banco prova a bassa potenza, è stato implementato un calcolo che rendesse possibile una misurazione indiretta della coppia. Riportiamo quindi di seguito l'equazione 2.5 che è stata utilizzata nei calcoli:

$$T = \frac{k_T}{R_i}(V - k_e\omega)$$

In cui V è la tensione letta nel motore quindi la coppia è stata calcolata tramite il lato motore del banco. Non è stato possibile invece usare tutte le equazioni riportanti la resistenza esterna (lato carico del banco) come ad esempio l'equazione 2.11:

$$T = \frac{k_T}{R_e + 2R_i}V$$

e neanche l'equazione derivata dalla conservazione di potenza

$$P = T\omega$$

$$P = VI = \frac{V^2}{R_e}$$

per cui si può definire la coppia come:

$$T = \frac{V^2}{R_e\omega} \quad (5.1)$$

Non si possono eseguire calcoli in cui viene inclusa la presenza di una resistenza esterna in quanto ciò è valido esclusivamente quando il PWM comandato sul carico ha valore massimo, ovvero quando il circuito è completamente chiuso. In caso contrario, con un duty cycle sul transistor inferiore al 100%, vi sono dei periodi di

tempo, seppur molto limitati, in cui fisicamente la resistenza esterna viene esclusa dal circuito e le equazioni non possono essere usate.

Un altro passo utile al controllo della coppia è la taratura del valore di resistenza di potenza da usare nel circuito. Siccome la corrente massima del motore è 2,7 A è necessario che la corrente prodotta nella resistenza sia minore:

$$I = \frac{E_{max}}{R_i + R_e} < I_{max}$$

in cui:

- $R_i = 4,5 \Omega$
- $E_{max} = 24 \text{ V}$
- $I_{max} = 2,7 \text{ A}$

Rispettivamente si trova:

$$R_e > \frac{E_{max} - R_i I_{max}}{I_{max}} = 4,40 \Omega \quad (5.2)$$

Inoltre non può essere usata una resistenza di potenza troppo grande perché in tal caso si avrebbe una corrente indotta troppo bassa e conseguentemente anche la coppia ottenuta sarebbe trascurabile. Infatti dall'equazione 5.1 si evince che più è alta la resistenza esterna e meno effetti si avranno sulla coppia. Quindi si può concludere che la resistenza esterna R_e debba essere minore di 10 Ω .

Un altro parametro fondamentale riguardo la scelta della resistenza consiste nella potenza che esse riescono a dissipare: dato che il motore ha un massimo di potenza di 40 W la potenza dissipata dalla resistenza dovrà essere maggiore o uguale a tale valore.

Nel caso progettato, si avevano in dotazione tre resistenze da circa 1 Ω ciascuna, la cui somma di potenza dissipata è 40 W. Mettendo in parallelo le resistenze, la resistenza equivalente, calcolata tramite multimetro, è 0.8 Ω ; questo è un valore troppo alto. Mettendo invece le tre resistenze in serie si ottiene 3.5 Ω . Un valore soddisfacente ma che limita la tensione massima ad un valore inferiore di 24 V, anche per questo motivo verrà usata una tensione massima di 17 V nel testing.

5.1 Circuito usato per il controllo della coppia

In fig. 5.4 si può trovare il circuito finale realizzato per il controllo della coppia. La scelta dei transistor viene qua riportata: è possibile sia scegliere di realizzare un collegamento manuale con il transistor oppure comprare un modello di transistor già preimpostato.

Nel primo caso la scelta è ricaduta su un transistor MOSFET IRL530N con le seguenti caratteristiche: potenza massima dissipata di 79 W, tensione massima di 100 V e corrente massima di 17 A. Il transistor, il cui simbolo viene riportato in fig. 5.1, in questo caso deve essere inserito nel circuito con i tre piedini collegati nel seguente modo:

- Pin 1 - gate : è collegato al Pin 9 di Arduino (usato come output)
- Pin 1 - drain : è collegato al motore
- Pin 1 - source : è collegato alla massa

Questo transistor è stato scelto per il controllo della tensione in ingresso al motore. Per evitare il ricircolo delle sovracorrenti è necessario un diodo collegato in parallelo al motore. I diodi in dotazione sono 1N4007: si tratta di diodi rettificatori in silicio per applicazioni a bassa potenza con corrente di passaggio fino ad 1 A. Siccome la corrente circolante nel motore è variabile fino a 2.7 A sono stati inseriti tre diodi.

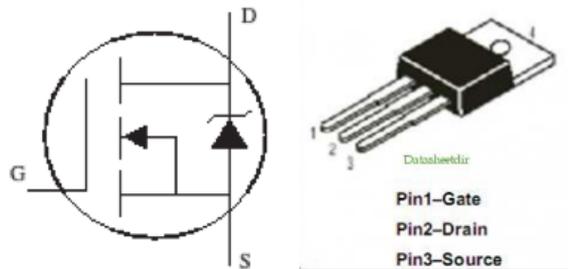


Figura 5.1: Simbolo MOSFET IRL530N

Nel secondo caso si può usare un modulo preimpostato con diodo incluso: si tratta del Modulo IRF520, in fig. 5.2, con cui si può comandare un carico fino a 30 V in corrente continua e corrente massima di 5 A. Si tratta di un collegamento più comodo che è stato scelto per il controllo del carico.

In entrambi i casi ai transistor vengono collegate delle alette dissipatrici, di cui necessitano da datasheet.

Sulla Breadboard di Arduino sono presenti collegamenti per la lettura della tensione nel carico e nel motore: servono dei partitori di tensione (in questo caso sono formati da resistenze di 2 e 10 Ω e da un diodo rettificatore 1N4007) poiché

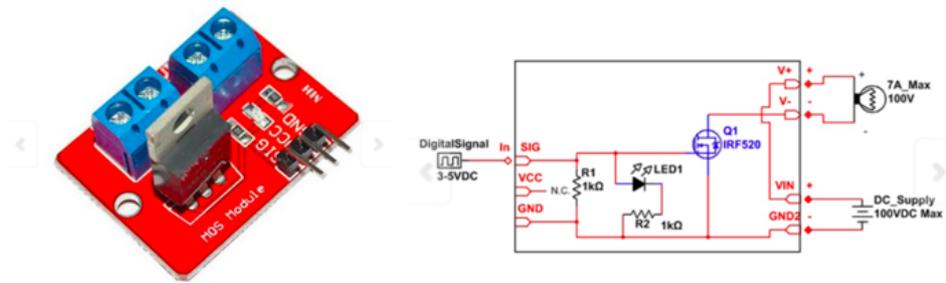


Figura 5.2: Modulo MOSFET IRF520

Arduino supporta un massimo di 5 V. Viene anche inserito un condensatore da 100 μF per avere una lettura della tensione più costante.

Per la lettura della corrente si sono usati due metodi. La corrente nel motore è stata calcolata tramite l'equazione 2.1 sostituendo alla tensione V quella letta sperimentalmente:

$$I_{motore} = \frac{1}{R_i}(V_{motore} - k_e\omega) \quad (5.3)$$

Diversamente, siccome la tensione letta nel carico è un valore più oscillante è stato inserito un sensore di corrente, collegato in serie alla resistenze. Il sensore in fig. 5.3 è il ACS712 (5 AMP) ed è grado di rilevare con precisione la corrente AC o DC fino a 5 Ampere.

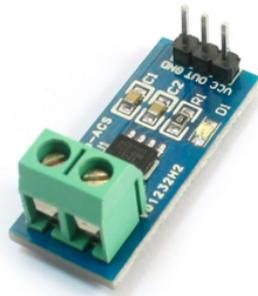


Figura 5.3: Sensore di corrente ACS712 (5 AMP)

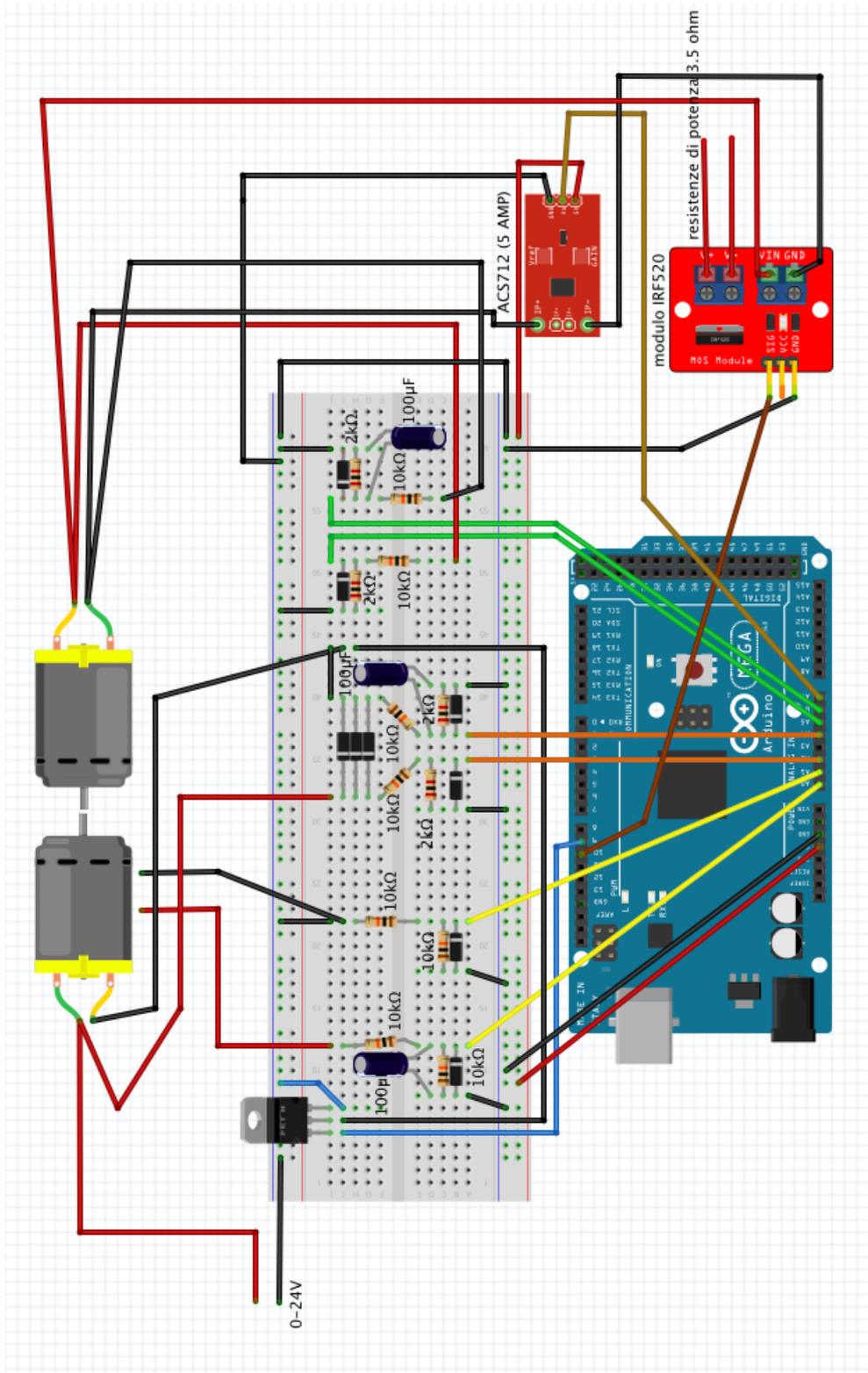


Figura 5.4: Circuito di controllo della coppia

5.2 Controllo del carico ad anello aperto

Il codice usato al fine di controllare la coppia con sistema ad anello aperto è riportato in fig. 5.5 e in fig. 5.6. Il sistema ha come parametri di input il PWM impostato sul carico (carico massimo = 255/255, carico nullo = 0/255) e la tensione da fornire al motore. Il sistema comunica il dutycycle al transistor collegato alla resistenza esterna e inizia a circolare corrente dalla parte del carico. Si crea quindi una coppia resistente che viene calcolata con l'equazione 2.5. Inoltre tramite questo codice si posso trovare sperimentalmente anche gli andamenti delle tensioni e delle correnti circolanti sia nel motore che nel carico. E' anche interessante studiare il plot della velocità rispetto alla variazione del carico.

Il controllo della coppia ad anello aperto consiste nel comunicare al transistor collegato alla resistenza esterna un dutycycle proporzionale alla coppia voluta, secondo l'equazione:

$$PWM_{coppia} = \frac{coppia_{voluta}[Nmm]}{coppia_{massima}[Nmm]} * 255 \quad (5.4)$$

con

$$coppia_{massima}[Nmm] = 1000 * k_T * \frac{V_{applicata}[V]}{(R_i + R_e)[\Omega]}$$

Il controllo ad anello aperto, di cui l'equazione 5.4 non tiene conto dei disturbi (es: presenza di una coppia, di resistenze, attriti, cadute di tensione sull'alimentazione, ecc..).

```
#include <PID_v1.h>
#define dinamoPOS A0
#define dinamoNEG A1
#define LetturaTensioneMotorePOS A2
#define LetturaTensioneMotoreNEG A3
#define motore 9
#define carico 10
#define LetturaTensioneCaricoPOS A4
#define LetturaTensioneCaricoNEG A5
#define LetturaCorrenteCarico A6

//valori che si possono cambiare
double pwmCarico =0; // 0-255 coppia da impostare
float tensioneApplicata = 0; // velocità da impostare
float tensioneTrasformatore = 17;
//
```

Figura 5.5: Codice di controllo del carico ad anello aperto (prima parte)

```

volatile int tensioneDinamoPOS = 0;
volatile int tensioneDinamoNEG = 0;
volatile int TensioneMotoreLettaPOS, TensioneMotoreLettaNEG;
volatile int TensioneCaricoLettaPOS, TensioneCaricoLettaNEG;
volatile int CorrenteCaricoLetta;
float coppia;
float velocita;
unsigned long t1;
int pwmMotore = 0;
float Samples=0.0, CorrenteCaricoLettaMedia=0.0;
int x=0;
float CorrenteCarico=0;
float resistenzaInternaMotore = 4.5; //ohm

void setup() {
  TCCR1B = TCCR1B & 0b11111000 | 0x02;

  Serial.begin(9600);
  pinMode(dinamoPOS, INPUT);
  pinMode(dinamoNEG, INPUT);
  pinMode(LetturaTensioneMotorePOS, INPUT);
  pinMode(LetturaTensioneMotoreNEG, INPUT);
  pinMode(LetturaTensioneCaricoPOS, INPUT);
  pinMode(LetturaTensioneCaricoNEG, INPUT);
  pinMode(LetturaCorrenteCarico, INPUT);
  pinMode(motore, OUTPUT);
  pinMode(carico, OUTPUT);
  digitalWrite(motore, HIGH);
}
void loop() {
  pwmMotore = tensioneApplicata*255/tensioneTrasformatore;
  analogWrite(carico, pwmCarico);
  analogWrite(motore, pwmMotore);

  //ALIMENTAZIONE
  TensioneMotoreLettaPOS = analogRead(LetturaTensioneMotorePOS);
  TensioneMotoreLettaNEG = analogRead(LetturaTensioneMotoreNEG);

  float TensioneMotoreLettaVolt = (TensioneMotoreLettaPOS - TensioneMotoreLettaNEG)*30/1023.0; //V
  // Serial.print("Tensione_di_alimentazione_del_motore:\t");
  // Serial.println(TensioneMotoreLettaVolt);
  float CorrenteMotore = (TensioneMotoreLettaVolt-8*velocita/1000.0)/resistenzaInternaMotore; //A
  // Serial.print("Corrente_di_alimentazione_del_motore:\t");
  // Serial.println(CorrenteMotore);
}

```

Figura 5.6: Codice di controllo del carico ad anello aperto (seconda parte)

```

//CARICO
TensioneCaricoLettaPOS = analogRead(LetturaTensioneCaricoPOS);
TensioneCaricoLettaNEG = analogRead(LetturaTensioneCaricoNEG);

float TensioneCaricoLettaVolt = (TensioneCaricoLettaPOS - TensioneCaricoLettaNEG)*30/1023.0;
//      Serial.print("Tensione_che_passa_nel_carico:\t");
//      Serial.println(TensioneCaricoLettaVolt);

if (x<100) {
    CorrenteCaricoLetta = analogRead(LetturaCorrenteCarico);
    Samples = Samples + CorrenteCaricoLetta;
    x++;
}

else{
    x=0;
    CorrenteCaricoLettaMedia=Samples/100;
    CorrenteCarico = (2.5 - (CorrenteCaricoLettaMedia*(5.0/1024.0)))/0.185;
//      Serial.print("Corrente_che_passa_nel_carico:\t");
//      Serial.println(CorrenteCarico);
    Samples=0;
}

//
//DINAMO
tensioneDinamoPOS = analogRead(dinamoPOS); //numero tra 0 e 1023
tensioneDinamoNEG = analogRead(dinamoNEG);
float tensioneDinamoVolt = (tensioneDinamoPOS - tensioneDinamoNEG)*(10/1023.0);
//      Serial.print("tensione_della_dinamo_tachimetrica:\t");
//      Serial.println(tensioneDinamoVolt);

velocita = (tensioneDinamoPOS - tensioneDinamoNEG)*(1000/3.0)*(10/1023.0); //rpm
//      Serial.print("Velocità:\t");
//      Serial.println(velocita); // rpm

//COPPIA
coppia = 1000.0*((0.0764/4.5)*TensioneMotoreLettaVolt-(0.0764*8.0/4.5)*velocita/1000.0); //mNm
//      Serial.print("Coppia:\t");
//      Serial.println(coppia); // rpm

```

Figura 5.7: Codice di controllo del carico ad anello aperto (terza parte)

5.3 Controllo del carico ad anello chiuso

In questo caso il controllo PID serve per comandare la coppia del sistema motore-carico. In input si ha la coppia ottenuta, calcolata tramite l'equazione 2.5, che viene confrontata con una coppia di riferimento impostata da chi vuole controllare. Il controllo PID quindi restituisce come output la tensione da sottoporre al carico per ottenere la coppia desiderata. Questa tensione serve per calcolare il dutycycle (ovvero il PWM) da comunicare al transistor. Il codice implementato è riportato in fig. 5.8 e in fig. 5.10

Il controllo di velocità ad anello chiuso somma il valore ottenuto con il controllo ad anello aperto, calcolato con l'equazione 5.4, con il risultato ottenuto dal controllo PID:

$$PWM_{coppia} = PWM_{outputPID} + \frac{coppia_{voluta}[Nmm]}{coppia_{massima}[Nmm]} * 255 \quad (5.5)$$

con

$$coppia_{massima}[Nmm] = 1000 * k_T * \frac{V_{applicata}[V]}{(R_i + R_e)[\Omega]}$$

In questa formulazione la parte del PID permette di compensare i disturbi, ma è legata alla precisione della dinamo tachimetrica e delle misure di tensioni.

In questo caso i parametri di PID ottimizzati sperimentalmente tramite il metodo di Ziegler-Nichols sono:

- pidtime: 0.01 ms
- K_p : 1.5
- K_i : 10
- K_d : 0.4

Inoltre è stata utilizzata una funzione per aumentare la frequenza del PWM di Arduino sui pin 9 e 10 che vengono usati come output nel nostro codice.

```
#include <PID_v1.h>
#define dinamoPOS A0
#define dinamoNEG A1
#define LetturaTensioneMotorePOS A2
#define LetturaTensioneMotoreNEG A3
#define motore 9
#define carico 10
#define LetturaTensioneCaricoPOS A4
#define LetturaTensioneCaricoNEG A5
#define LetturaCorrenteCarico A6
```

Figura 5.8: Codice di controllo del carico ad anello chiuso (prima parte)

```

//valori che si possono cambiare
double coppiaVoluta = 40; // [Nmm] coppia voluta da impostare tramite controllo PID
float tensioneApplicata =0; //[V] tensione impostata sul motore
double Kp=1.5, Ki=10, Kd=0.4;
unsigned long pidtime = 0.01;
float tensioneTrasformatore = 17;
//
volatile int tensioneDinamoPOS = 0;
volatile int tensioneDinamoNEG = 0;
volatile int TensioneMotoreLettaPOS,TensioneMotoreLettaNEG;
volatile int TensioneCaricoLettaPOS,TensioneCaricoLettaNEG;
volatile int CorrenteCaricoLetta;
double coppiaOttenuta, tensioneCarico;
float velocita;
float tensioneCaricoVolt = 0;
unsigned long t1;
int pwmCarico = 0;
float Samples=0.0, CorrenteCaricoLettaMedia=0.0;
int x=0;
float CorrenteCarico=0;

int resistenzaPartitore_motore = 12; // kohm
int resistenzaPartitore_carico = 12;
float resistenzaCarico = 3.5; //ohm
float PotMaxNomMotore = 40; //W
float velMaxMotore = 3000; //rpm
float resistenzaInternaMotore = 4.5; //ohm

float correnteMax = (tensioneApplicata-3)/(resistenzaInternaMotore+resistenzaCarico);
float coppiaMax = 1000*0.0764*correnteMax;
int pwmMotore = tensioneApplicata*255/tensioneTrasformatore;

PID myPID(&coppiaOttenuta, &tensioneCarico, &coppiaVoluta, Kp, Ki, Kd, DIRECT);

void setup() {

  TCCR1B = TCCR1B & 0b11111000 | 0x02;

  Serial.begin(9600);
  pinMode(dinamoPOS,INPUT);
  pinMode(dinamoNEG,INPUT);
  pinMode(LetturaTensioneMotorePOS,INPUT);
  pinMode(LetturaTensioneMotoreNEG,INPUT);
  pinMode(LetturaTensioneCaricoPOS,INPUT);
  pinMode(LetturaTensioneCaricoNEG,INPUT);
  pinMode(LetturaCorrenteCarico,INPUT);
  pinMode(motore,OUTPUT);
  pinMode(carico,OUTPUT);
  digitalWrite(motore, HIGH);

  myPID.SetSampleTime(pidtime);
  myPID.SetOutputLimits(0, 255);
  myPID.SetMode(AUTOMATIC);
}

```

Figura 5.9: Codice di controllo del carico ad anello chiuso (seconda parte)

```

void loop() {

  analogWrite(carico, pwmCarico);
  analogWrite(motore, pwmMotore);

  if ((millis() - t1) >= pidtime) {

    //ALIMENTAZIONE
    TensioneMotoreLettaPOS = analogRead(LetturaTensioneMotorePOS);
    TensioneMotoreLettaNEG = analogRead(LetturaTensioneMotoreNEG);

    float TensioneMotoreLettaVolt = (TensioneMotoreLettaPOS - TensioneMotoreLettaNEG)*30/1023.0;
    // Serial.print("Tensione_di_alimentazione_del_motore:\t");
    // Serial.println(TensioneMotoreLettaVolt);
    float CorrenteMotore = (TensioneMotoreLettaVolt-8*velocita/1000.0)/resistenzaInternaMotore; //mA
    // Serial.print("Corrente_di_alimentazione_del_motore:\t");
    // Serial.println(CorrenteMotore);

    //CARICO
    TensioneCaricoLettaPOS = analogRead(LetturaTensioneCaricoPOS);
    TensioneCaricoLettaNEG = analogRead(LetturaTensioneCaricoNEG);

    float TensioneCaricoLettaVolt = (TensioneCaricoLettaPOS - TensioneCaricoLettaNEG)*30/1023.0;
    // Serial.print("Tensione_che_passa_nel_carico:\t");
    // Serial.println(TensioneCaricoLettaVolt);

    if (x<150) {
      CorrenteCaricoLetta = analogRead(LetturaCorrenteCarico);
      Samples = Samples + CorrenteCaricoLetta;
      x++;
    }
    else{
      x=0;
      CorrenteCaricoLettaMedia=Samples/100;
      CorrenteCarico = (2.5 - (CorrenteCaricoLettaMedia*(5.0/1024.0)))/0.185;
      // Serial.print("Corrente_che_passa_nel_carico:\t");
      // Serial.println(CorrenteCarico);
      Samples=0;
    }

    //DINAMO
    tensioneDinamoPOS = analogRead(dinamoPOS); //numero tra 0 e 1023
    tensioneDinamoNEG = analogRead(dinamoNEG);
    float tensioneDinamoVolt = (tensioneDinamoPOS - tensioneDinamoNEG)*(10/1023.0);
    // Serial.print("tensione_della_dinamo_tachimetrica:\t");
    // Serial.println(tensioneDinamoVolt);
  }
}

```

Figura 5.10: Codice di controllo del carico ad anello chiuso (terza parte)

Capitolo 6

Risultati sperimentali del controllo di velocità

Nel seguente capitolo sono riportati tutti i grafici ottenuti dalle sperimentazioni. Si possono quindi trovare sia diversi controlli ad anello aperto (o open-loop) che diversi controlli PID ad anello chiuso (o close-loop). Inoltre nel controllo PID sono stati testati diversi profili di velocità, in particolare: un profilo a step, uno a forma sinusoidale e uno a trapezio.

6.1 Grafici del controllo di velocità ad anello aperto

In fig. 6.1 sono state testate diverse tensioni di alimentazione del motore tramite il controllo PWM su un transistor e sono state evidenziate le seguenti velocità massime:

- tensione 17 V : 2046 rpm
- tensione 15 V : 1756 rpm
- tensione 10 V : 1707 rpm
- tensione 5 V : 130 rpm

Si nota che alle tensioni più alte, ovvero a PWM maggiori, la variazione di velocità è minore. Tra 10 e 17 V si può notare che il transitorio inizialmente porta il sistema ad una velocità di circa 1780 rpm. Ciò significa che il sistema è improntato a reagire alle variazioni di tensione di alimentazione portando il banco prova a girare inizialmente ad una certa velocità per poi più lentamente raggiungere la velocità attesa, corrispondente alla tensione realmente impostata. Ovviamente questo non

riguarda le velocità più basse per cui si arriva direttamente alla velocità corrispettiva. In questo caso si nota sul banco prova che il sistema subisce degli scatti, i quali si evincono dal fattore di oscillazione del grafico. Ai valori inferiori di tensione si hanno velocità molto basse; questo è dovuto in parte agli attriti statici da vincere per far ruotare il sistema.

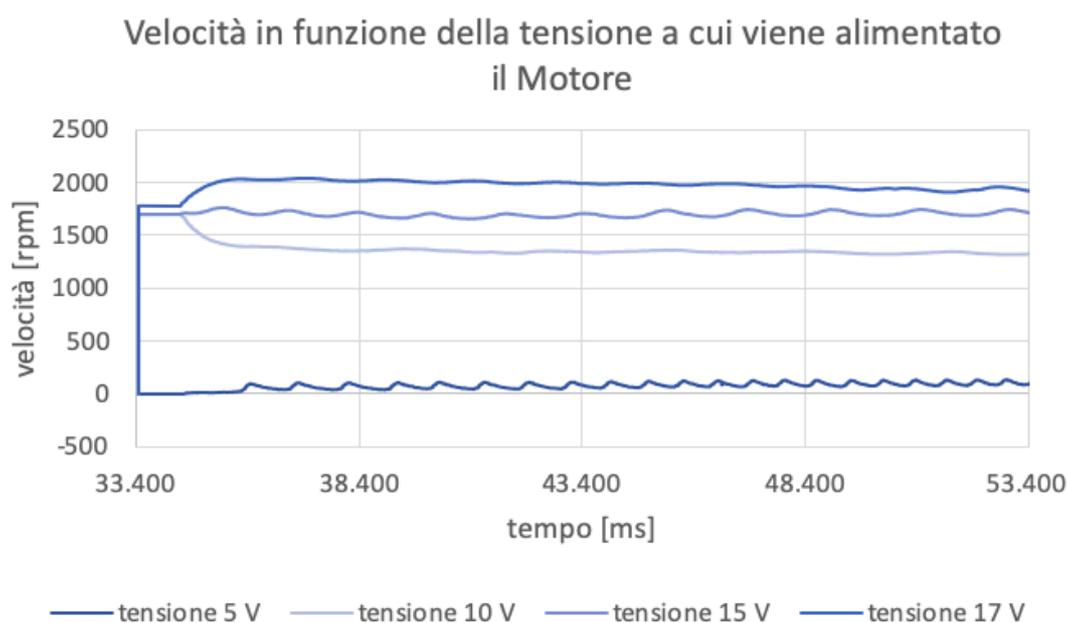


Figura 6.1: Velocità in funzione della tensione a cui viene alimentato il motore

In fig. 6.2 sono riportate le diverse tensioni lette nel motore a diversi PWM in entrata al transistor. Si può notare che la tensione reale letta nel motore non coincide mai con la tensione impostata con il PWM: ciò è dovuto a correnti indotte nel sistema che modificano temporaneamente la tensione di alimentazione massima. Inoltre è visibile come all'aumentare della tensione letta nel motore, e quindi del PWM applicato al transistor, diminuisca l'oscillazione del segnale.

In fig. 6.3 vengono riportate le correnti lette nel motore a diverse tensioni di alimentazione dello stesso. Si ha un andamento simile a quello del grafico di fig. 6.2 con un'aggiunta: infatti si può notare come il periodo delle oscillazioni sia proporzionale all'aumento del PWM.

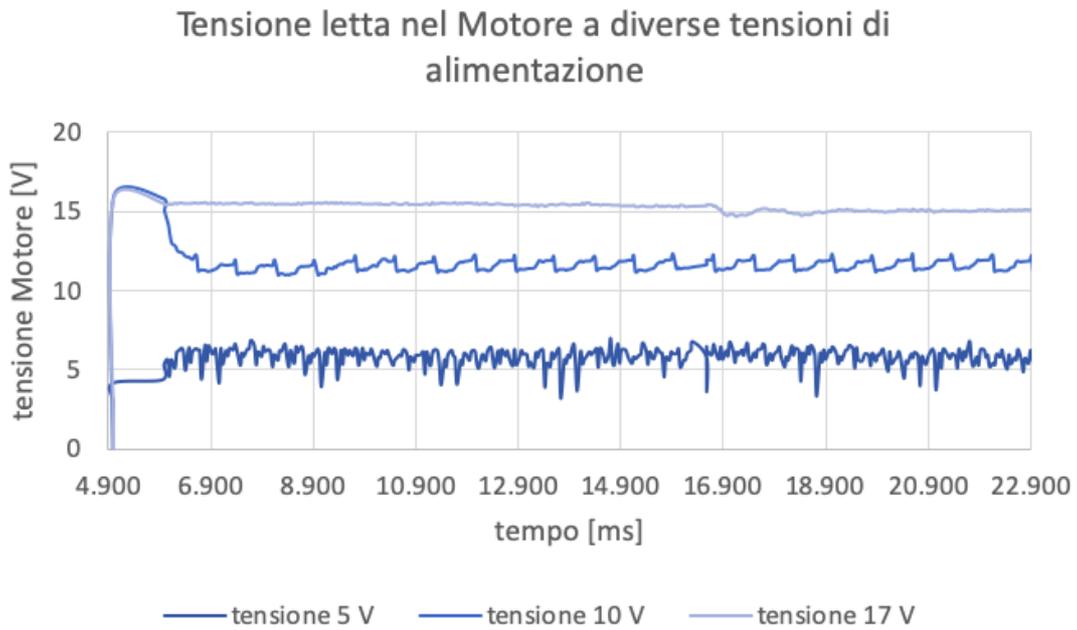


Figura 6.2: Tensione effettiva letta nel motore a diverse tensioni di alimentazione

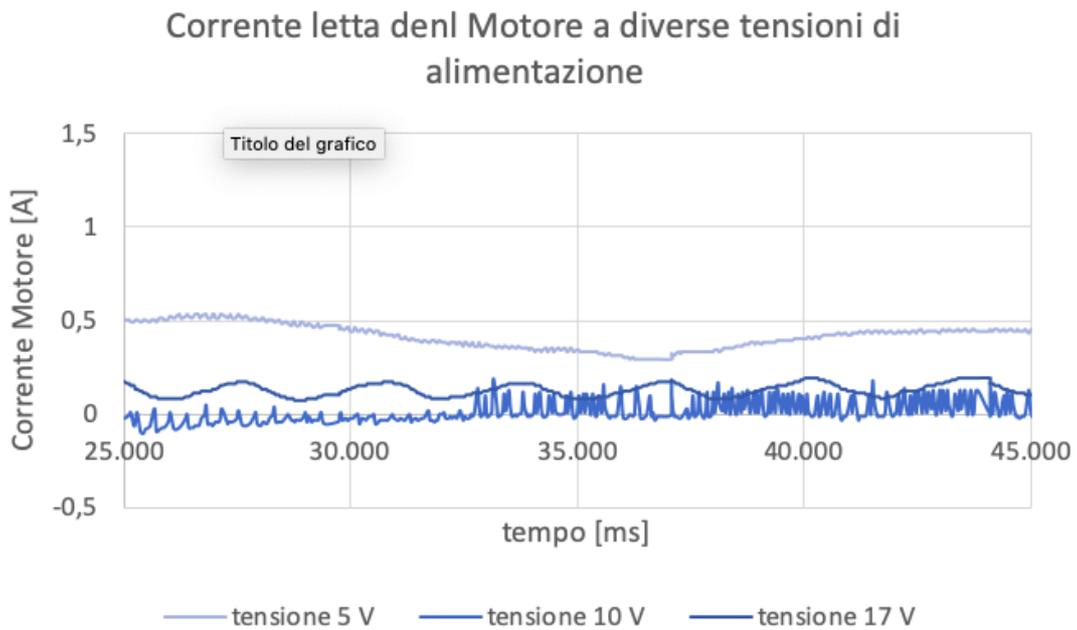


Figura 6.3: Corrente letta nel motore a diverse tensioni di alimentazione

6.2 Grafici del controllo PID di velocità (closed-loop)

In fig. 6.4 si possono trovare diverse velocità impostate per il controllo del banco prova: 1000 rpm, 1600 rpm e 1800 rpm. Il sistema è stato alimentato a massimo 17 V e il PWM, impostato sul transistor collegato al motore, è quindi calcolato su un massimo di 17 V. In particolare in questo grafico si può notare che i parametri del PID, ovvero

$$K_p = 0.01, K_i = 0.2, K_d = 0.001$$

rendono il PID regolare, con:

- una sovralongazione massima più alta nel caso a bassa velocità
- un transitorio che aumenta con il crescere della velocità (ma comunque di pochi millisecondi)
- un errore statico molto basso che aumenta con la velocità

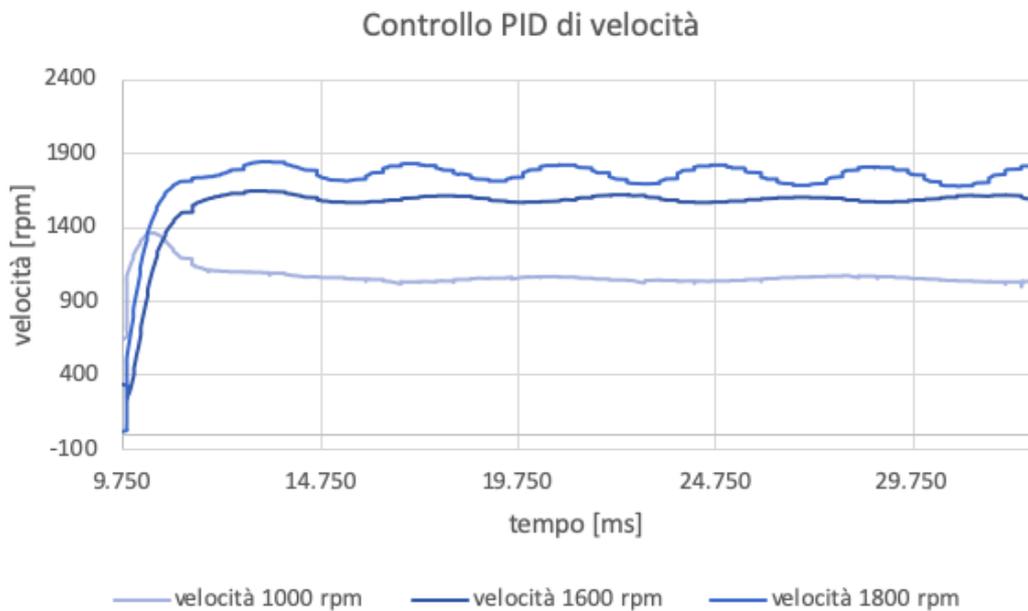


Figura 6.4: Diversi controlli PID di velocità

In fig. 6.5, si può notare come nel banco prova, sottoposto a diversi livelli di velocità, le tensioni lette nel motore sono poco oscillanti mentre oscillano maggiormente alle basse velocità, quando il PWM diminuisce.

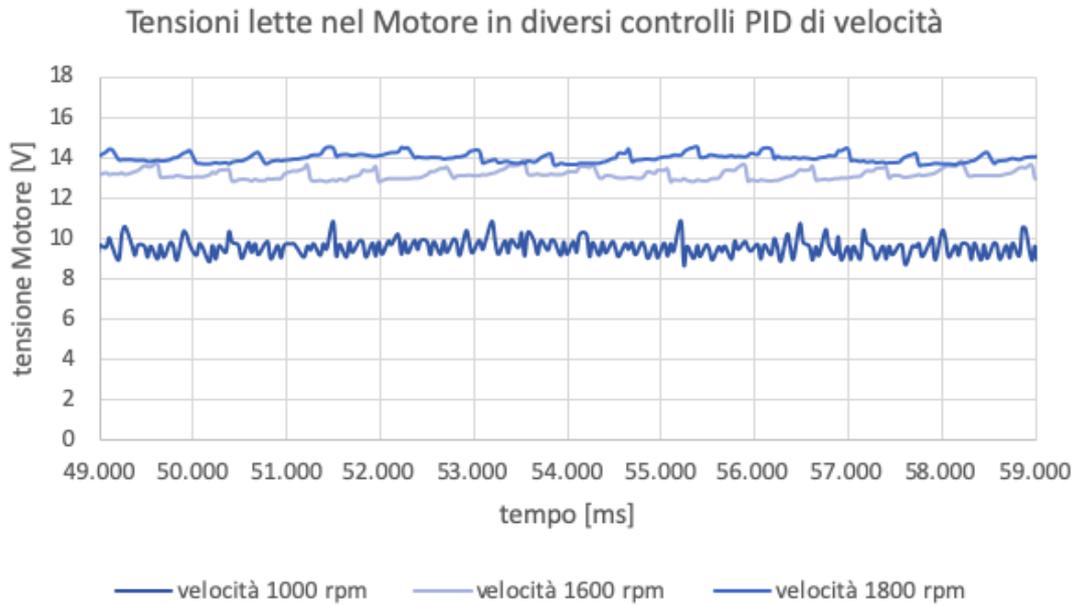


Figura 6.5: Tensioni lette in diversi controlli PID di velocità

Le correnti lette nel motore, il cui profilo si può osservare in fig. 6.6, invece sono visibilmente più oscillanti a velocità contenute, mentre a velocità elevate hanno un profilo più attenuato.

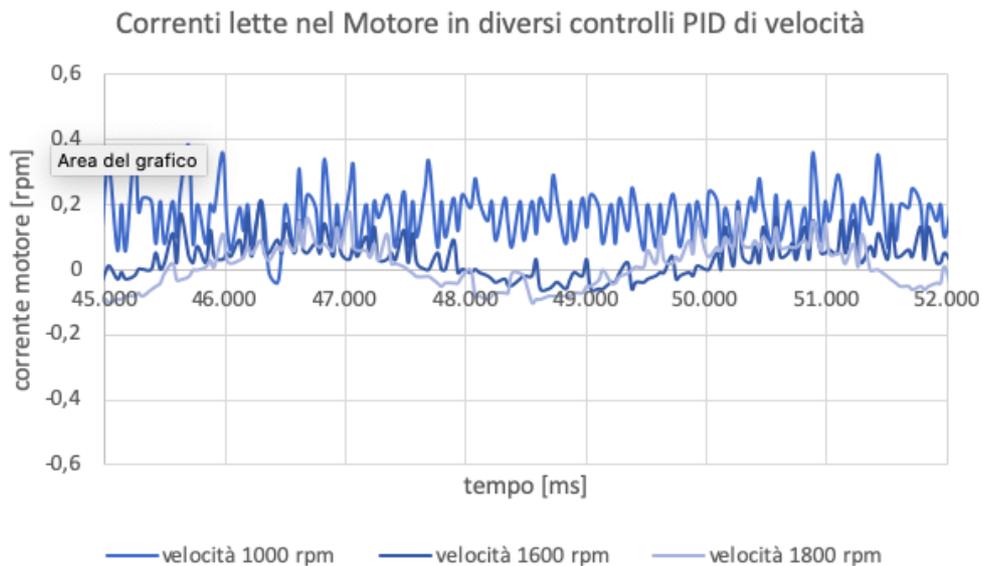


Figura 6.6: Correnti lette in diversi controlli PID di velocità

6.3 Grafici con effetto del carico sul controllo PID di velocità (closed-loop)

In questa sezione sono stati valutati gli effetti del controllo PID di velocità applicato al sistema in presenza di un carico massimo collegato al motore.

In fig. 6.7, si nota come in presenza di carico l'andamento del PID è privo di oscillazioni. Il transitorio è circa costante per tutte le velocità: inizialmente si ha un aumento della velocità molto rapido per poi arrivare al valore di regime in pochi millisecondi. Questo aumento così rapido è dovuto alla costante proporzionale, mentre la velocità con cui il sistema si stabilizza al valore di regime è dovuta all'azione integrativa.

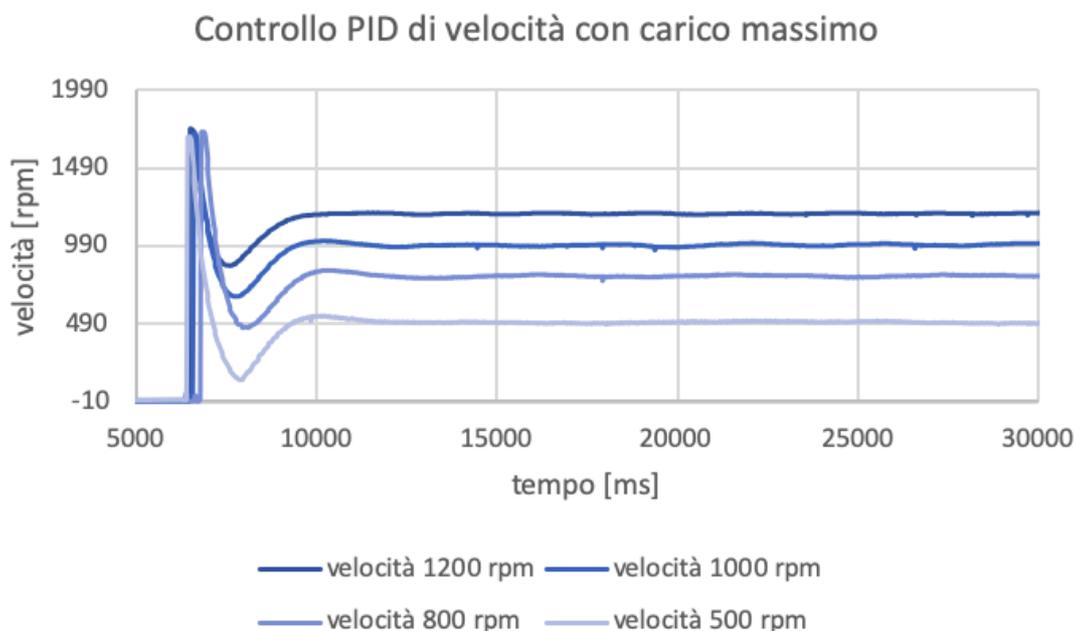


Figura 6.7: Diversi controlli PID di velocità con sistema collegato al carico massimo

In fig. 6.8, le oscillazioni delle tensioni lette nel motore sono proporzionali alla velocità di controllo del sistema. Questo è dovuto alla velocità di apertura e chiusura del transistor e quindi al valore di PWM a cui il sistema si stabilizza.

In fig. 6.9, in presenza di carico massimo collegato al sistema, le correnti lette nel motore sono quasi coincidenti. La differenza consiste nell'oscillazione che cresce proporzionalmente al diminuire della velocità di controllo.

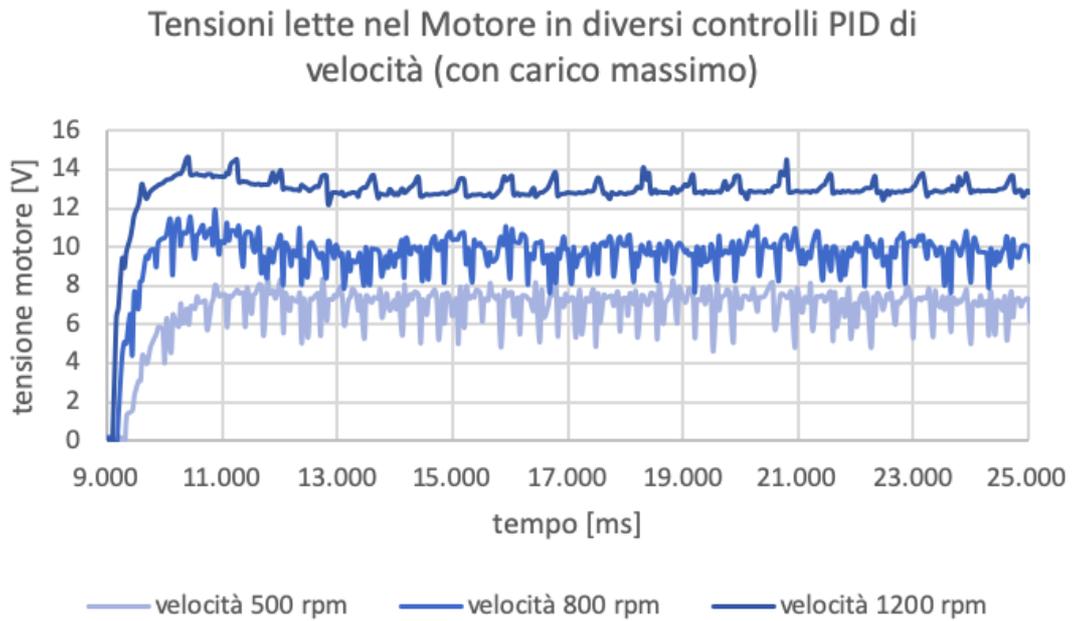


Figura 6.8: Tensioni lette in diversi controlli PID di velocità con sistema collegato al carico massimo

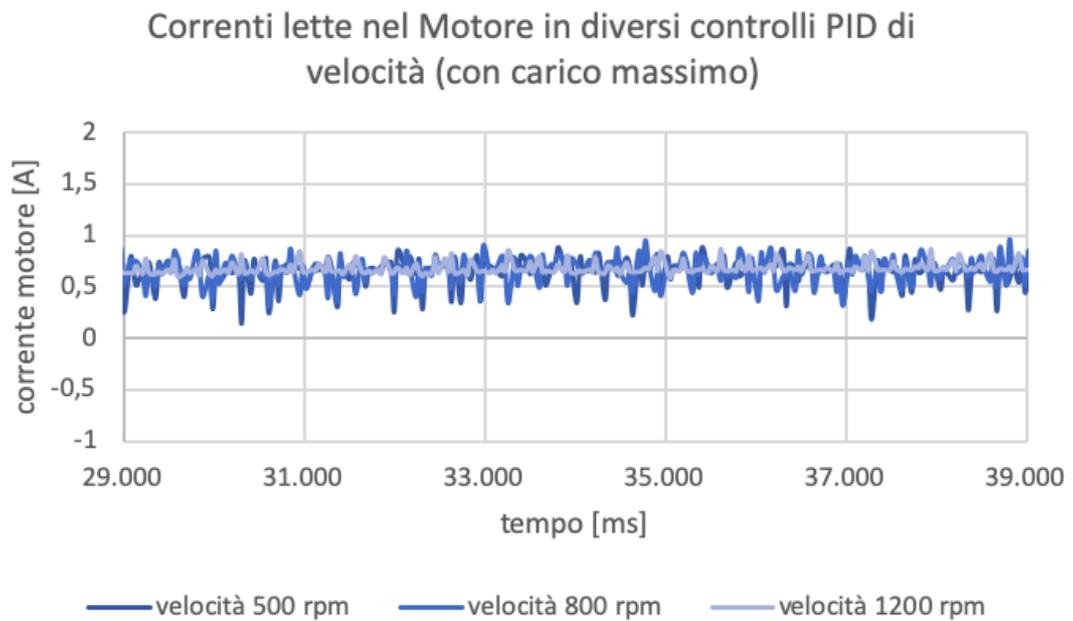


Figura 6.9: Correnti lette in diversi controlli PID di velocità con sistema collegato al carico massimo

6.4 Confronto dei grafici di controllo PID di velocità (closed-loop) in presenza e in assenza di carico

In questa sezione sono state confrontate le tensioni e le correnti lette sia nel motore, sia nel carico, nei due casi più estremi: in presenza di carico massimo e in assenza totale di carico.

6.4.1 Confronto tra le tensioni nel motore e nel carico

In fig. 6.10 e in fig. 6.11 sono rappresentate le tensioni lette nel motore e nel carico. La scelta delle velocità è stata dettata dalla presenza o dall'assenza del carico: il sistema in presenza di carico raggiunge in maniera più solida una velocità più bassa; inversamente in assenza di carico è possibile scegliere una velocità più alta. Le seguenti scelte sono state effettuate:

- velocità 1200 rpm: sistema in presenza di carico massimo
- velocità 1600 rpm: sistema in assenza di carico

La differenza sostanziale che si delinea dai grafici è il divario tra la tensione nel motore e la tensione nel carico: è maggiore nella condizione di carico massimo. Infatti in assenza di carico la riduzione di tensione tra motore e carico è dovuta solamente agli attriti interni del sistema, mentre in presenza di carico la riduzione è dovuta sia agli attriti che alla resistenza del carico aggiuntivo.

6.4.2 Confronto tra le correnti nel motore e nel carico

In fig. 6.12 e in fig. 6.13 si delineano le differenze tra le correnti lette nel motore e nel carico. Nel caso di applicazione del carico massimo le correnti hanno un piccolo divario in cui la corrente del motore è poco più grande della corrente che scorre nel carico. Nel caso in cui ci sia assenza di carico, le correnti quasi coincidono. In generale si nota che la corrente del motore è oscillante a causa del PWM comandato dal controllo PID sul motore, mentre la corrente nel carico è più stabilizzata in quanto si ha sulla dinamo freno in un caso PWM di 255/255, che quindi rappresenta un segnale costante dovuto al transistor completamente chiuso (carico massimo), e nell'altro caso si ha PWM di 0/255, che rappresenta sempre un segnale costante con transistor completamente aperto (assenza di carico).

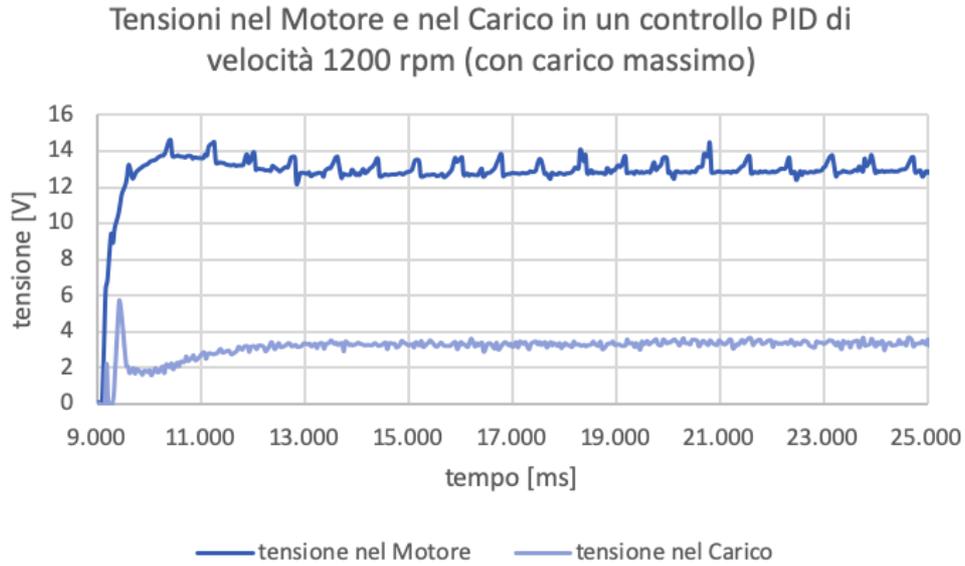


Figura 6.10: Confronto delle tensioni lette nel motore e nel carico in diversi controlli PID di velocità con sistema collegato al carico massimo

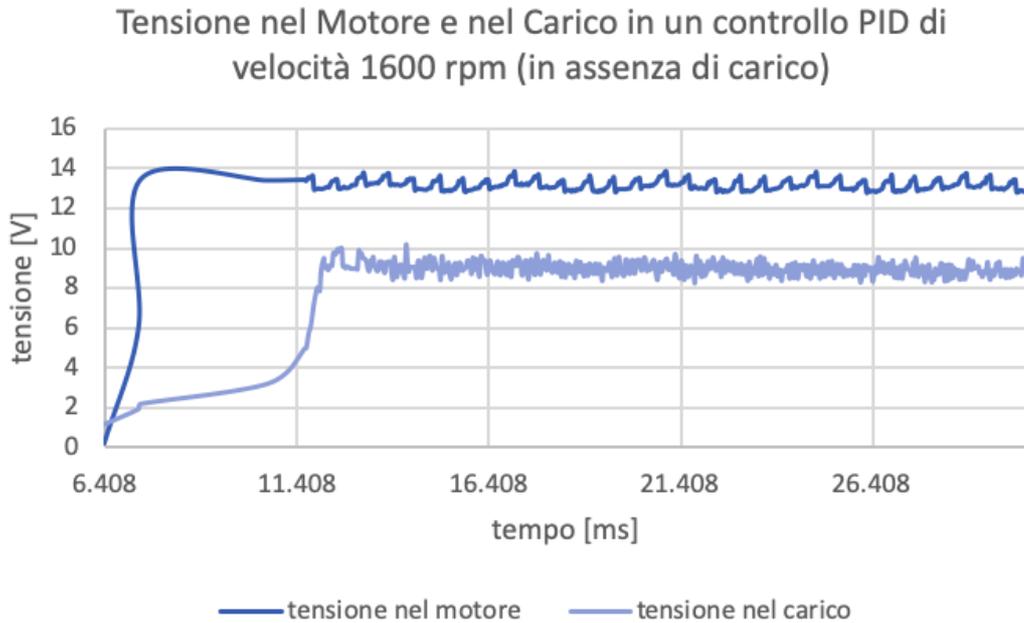


Figura 6.11: Confronto delle tensioni lette nel motore e nel carico in diversi controlli PID di velocità con sistema in assenza di carico

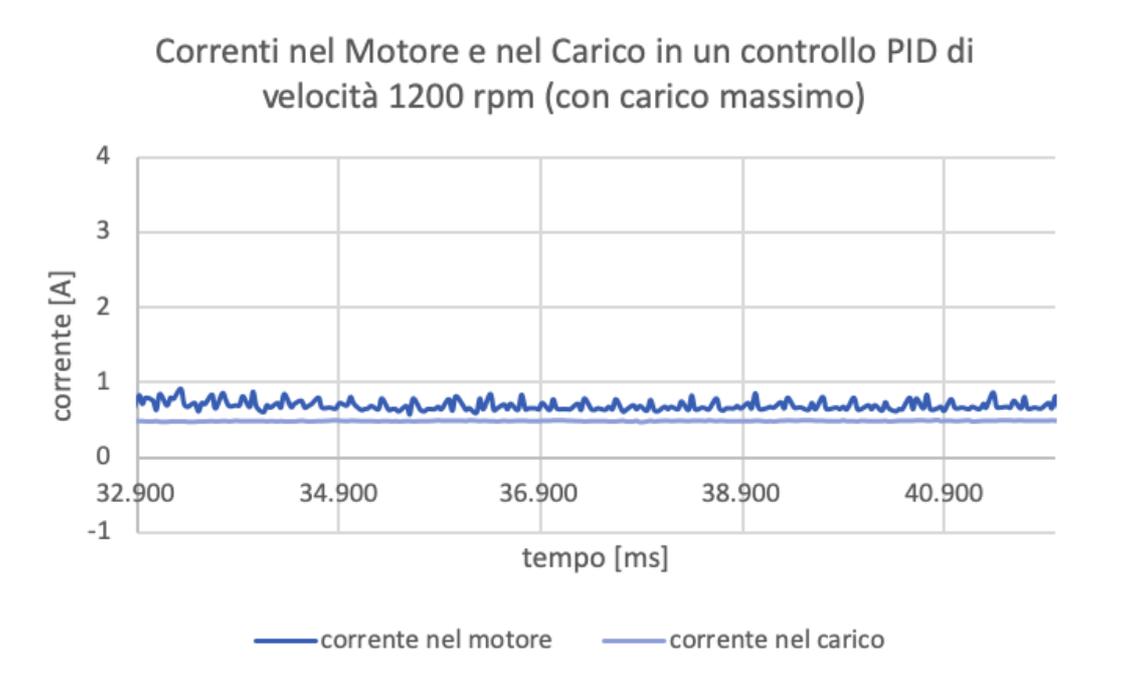


Figura 6.12: Confronto delle correnti lette nel motore e nel carico in diversi controlli PID di velocità con sistema collegato al carico massimo

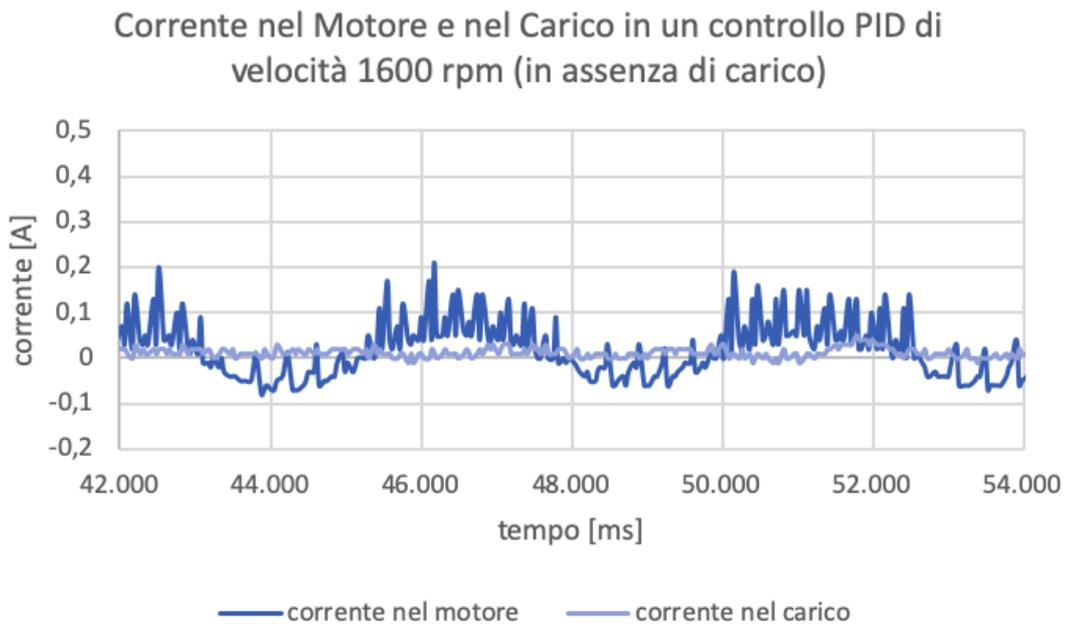


Figura 6.13: Confronto delle correnti lette nel motore e nel carico in diversi controlli PID di velocità con sistema in assenza di carico

6.5 Controllo PID di velocità con profilo a step

In questa sezione è stato testato un controllo PID con profilo di velocità a step. I parametri impostati per il profilo a step sono:

- prima velocità voluta: 1600 rpm
- seconda velocità voluta: 1400 rpm
- tempo di durata dello step: 5000 ms
- PWM impostato sul carico: 0/255

Il sistema è stato impostato con carico nullo in modo da poter testare velocità più elevate. Il controllo PID in fig. 6.14 evidenzia nel primo tratto una sovraelongazione massima maggiore dovuta sicuramente alla prima accelerazione del sistema che deve partire da fermo e arrivare ad una velocità elevata. Nei tratti seguenti il controllo PID ha quindi una sovraelongazione massima quasi nulla. Il controllo PID appare quindi soddisfacente siccome si ottiene anche un errore statico nullo.

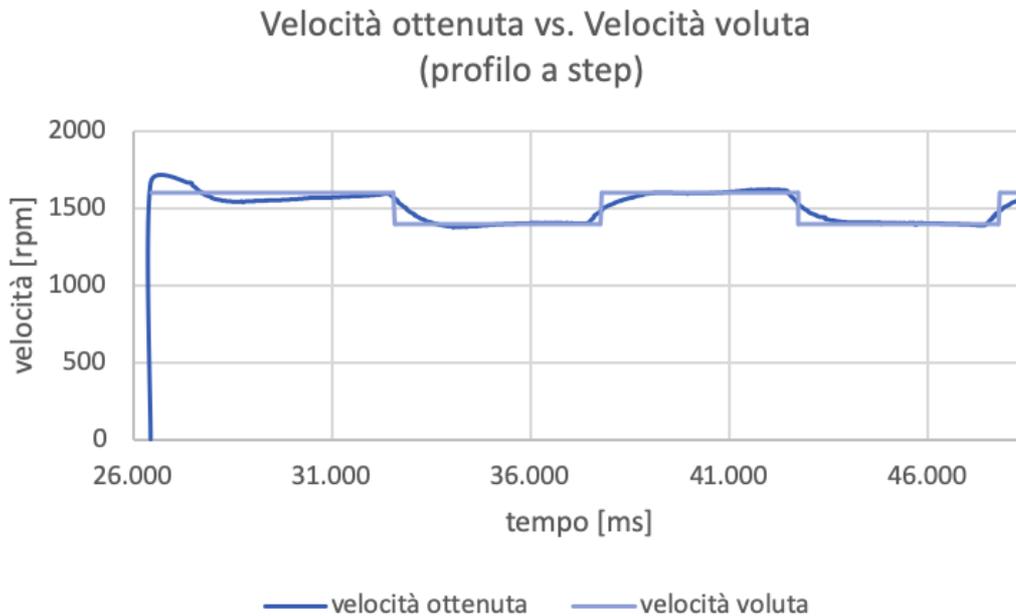


Figura 6.14: Controllo PID di velocità con profilo a step

In fig. 6.15 è stato plottato il confronto tra la tensione letta nel motore e quella letta nel carico: si nota che entrambe seguono lo stesso profilo, infatti aumentano e diminuiscono proporzionalmente alla velocità controllata dal PID. A causa degli attriti dell'accoppiamento e dell'efficienza elettrica dei motori, il sistema ha una

tensione maggiore nel motore rispetto a quella in uscita dal carico. Si può notare un piccolo ritardo di discesa e di salita nella tensione letta dal carico. Questo è dovuto all'inerzia del sistema.

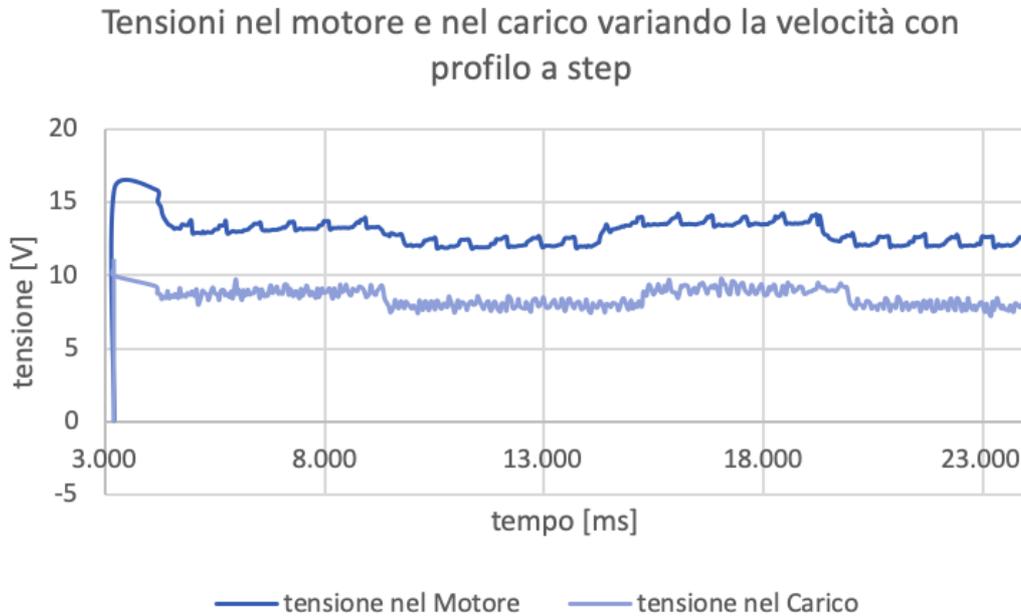


Figura 6.15: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo a step

In fig. 6.16 è presente il confronto tra corrente circolante nel motore e quella letta nel carico. La corrente che passa nel carico è quasi nulla, in quanto il circuito dalla parte del carico è completamente aperto (PWM pari a zero). La corrente nel motore è oscillante poiché è comandata da un PWM intermedio, ma rimane più o meno costante durante i cambi di velocità. Si possono giusto notare dei piccoli transitori i cui la corrente scende leggermente, ma si tratta di una situazione solo temporanea.

La coppia rappresentata in fig. 6.17 è indipendente dall'andamento del controllo di velocità poiché non dipende da una resistenza esterna (il sistema è in condizione di circuito completamente aperto dalla parte del carico) ma soltanto dall'inerzia del sistema e dalle perdite subite dall'accoppiamento. Si possono notare delle piccole variazioni che sono effettivamente dipendenti dalla velocità.

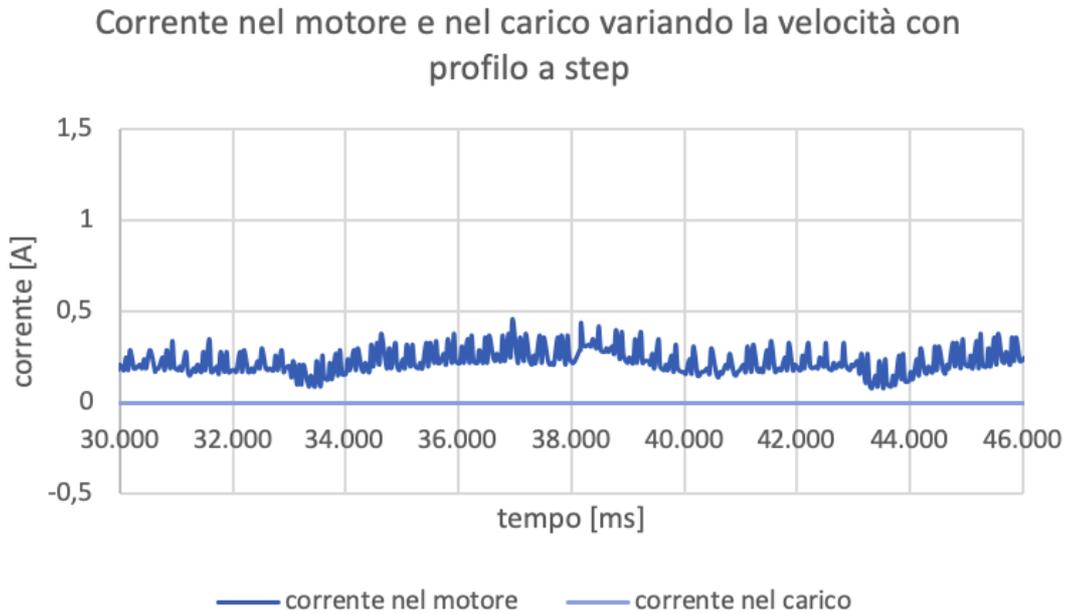


Figura 6.16: Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo a step

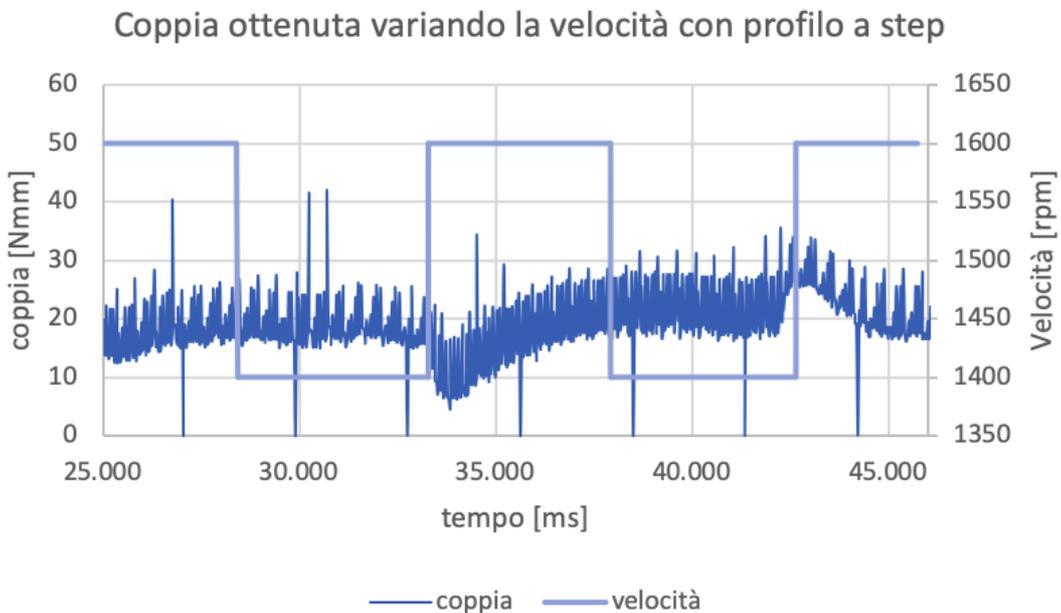


Figura 6.17: Coppia in un controllo PID di velocità con profilo a step

6.6 Controllo PID di velocità con profilo sinusoidale

In questo caso sono stati usati i seguenti parametri:

- Ampiezza del profilo sinusoidale di velocità: 1000 rpm
- Periodo del profilo sinusoidale: 5000 ms
- PWM impostato sul carico: 255/255.

In fig. 6.18 è stato plottato il grafico della velocità controllata dal PID rispetto alla velocità voluta dal profilo sinusoidale. Si nota un piccolo ritardo nella risposta del sistema, dovuto alla sua inerzia.

Il PWM massimo non permette di scegliere una velocità elevata perchè il sistema fa più fatica a vincere la coppia resistente.

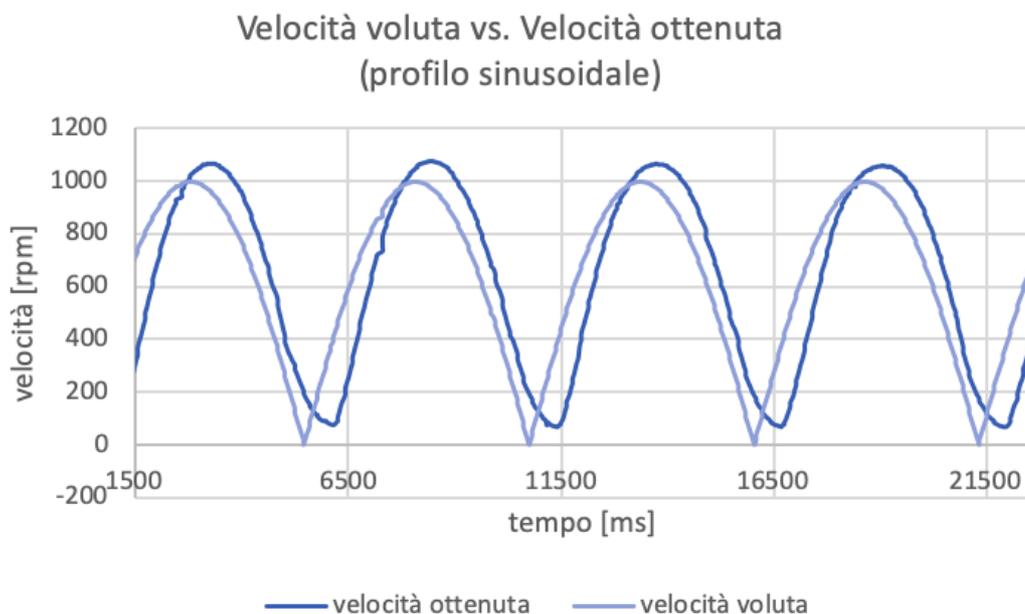


Figura 6.18: Controllo PID di velocità con profilo sinusoidale

In fig. 6.19, la tensione letta nel motore è proporzionale alla tensione letta nel carico. Entrambe seguono il profilo sinusoidale. La tensione nel motore è maggiore di quella nel carico a causa delle perdite che si hanno nell'accoppiamento.

In fig. 6.20, le correnti circolanti nel sistema hanno due andamenti simili infatti seguono il profilo sinusoidale comandato dal controllo di velocità. La corrente nel motore è leggermente più alta e più oscillante. Questo è dovuto dal fatto che il

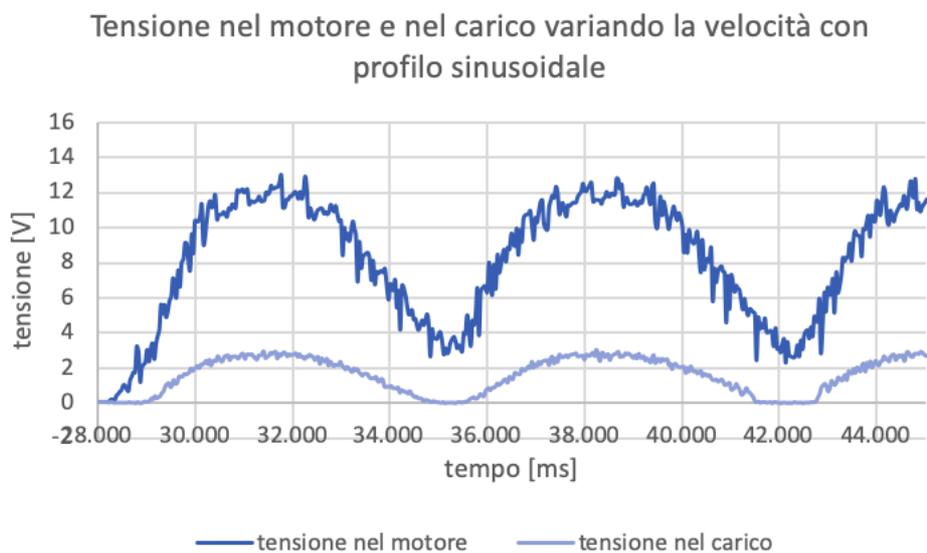


Figura 6.19: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo sinusoidale

PWM che controlla la tensione nel motore, e quindi la sua corrente, è un valore intermedio. Al contrario la corrente del carico è comandata da un PWM massimo e quindi risulta con meno oscillazioni.

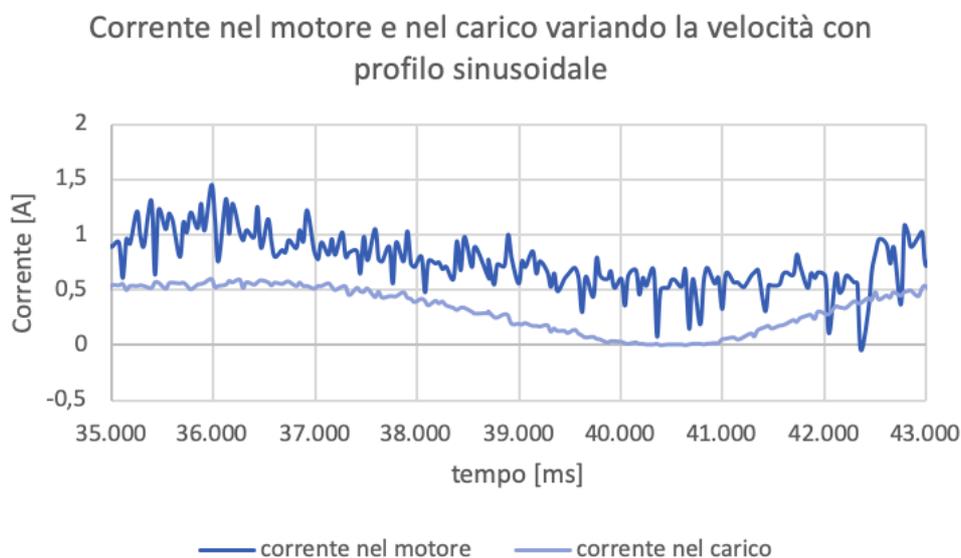


Figura 6.20: Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo sinusoidale

In fig. 6.21 si può vedere il comportamento della coppia. Siccome il sistema è in presenza di un carico, in questo caso è chiaramente più visibile l'andamento proporzionale alla velocità. La coppia ha un profilo sinusoidale, cresce all'aumentare della velocità controllata e viceversa. Ciò avviene poiché all'aumentare della velocità aumenta la tensione indotta sul carico e pertanto aumenta anche la corrente che vi circola, secondo la legge di Ohm. Siccome la coppia è proporzionale alla corrente risulta che un aumento della velocità produce un'aumento della coppia.

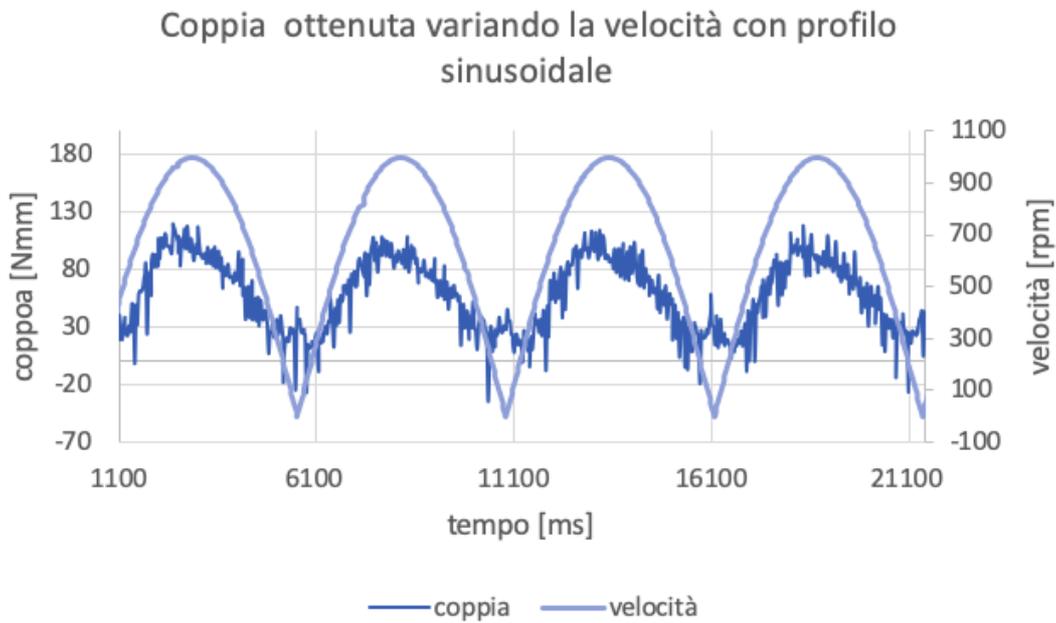


Figura 6.21: Coppia in un controllo PID di velocità con profilo sinusoidale

6.7 Controllo PID di velocità con profilo a trapezio

I parametri usati per simulare il profilo di velocità a trapezio sono:

- Velocità massima: 1400 rpm,
- Tempo di salita: 100 ms,
- Tempo a regime: 100 ms,
- Tempo di discesa: 100 ms,
- PWM applicato al carico: 0/255.

In fig. 6.22 si può trovare il comportamento del controllo PID di velocità con profilo di trapezio. Si può notare la presenza di overshoot ad ogni ciclo. Il sistema tende a stabilizzarsi e poi a seguire il profilo voluto con un minimo di ritardo. Ciò è dovuto al tempo in cui il segnale viene comunicato e all'inerzia del sistema.

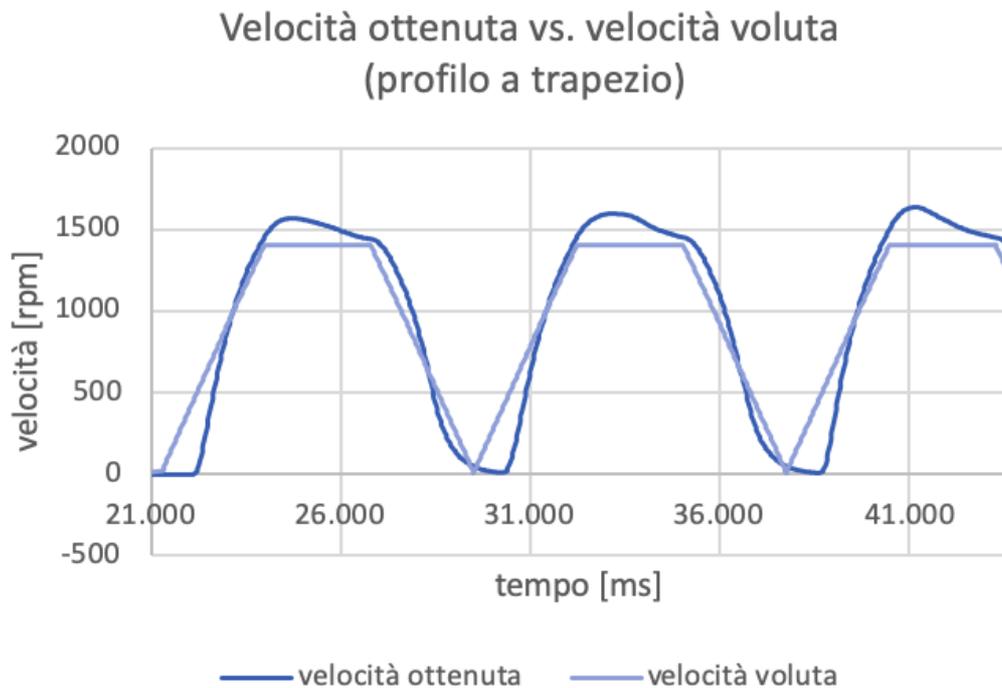


Figura 6.22: Controllo PID di velocità con profilo a trapezio

In fig. 6.23, le tensioni seguono il profilo trapezoidale del controllo PID. La tensione nel motore è più alta della tensione letta nel carico a causa delle perdite che si hanno nell'accoppiamento.

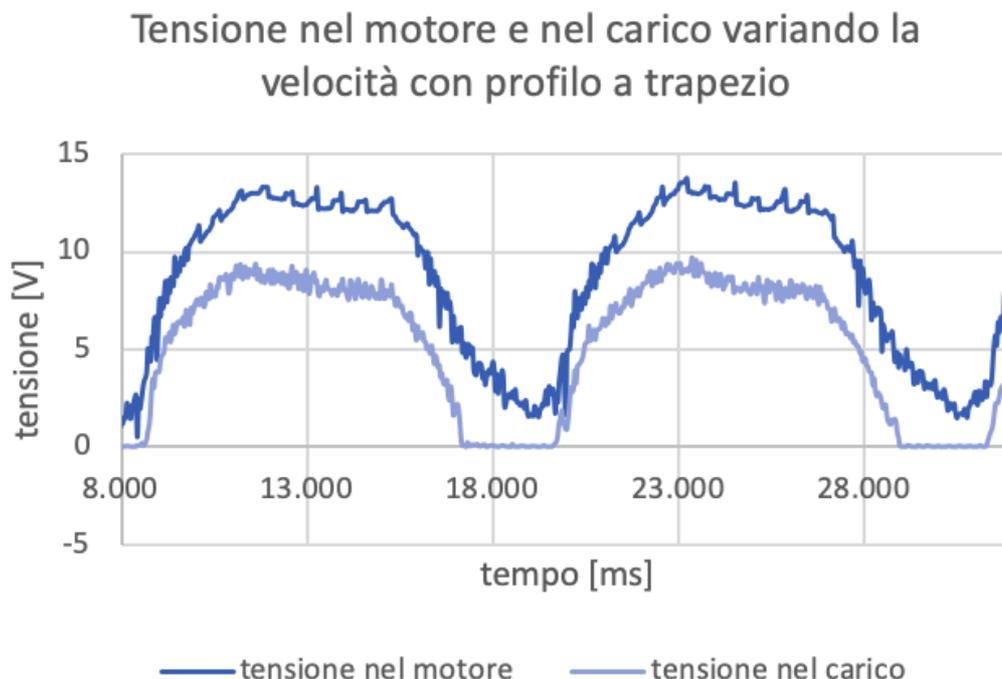


Figura 6.23: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di velocità con profilo a trapezio

In fig. 6.24 sono presenti le correnti lette nel sistema:

- la corrente nel carico è nulla in quanto il circuito collegato al carico è completamente aperto (PWM uguale a 0),
- la corrente nel motore ha dei picchi di accelerazione quando il sistema aumenta la velocità, per poi diminuire quando la velocità va mantenuta costante o diminuita.

In fig. 6.25 è rappresentato l'andamento della coppia in funzione del controllo PID di velocità; il profilo della coppia è quasi proporzionale a quello della velocità, ma ciò non è ben definito poichè l'unica componente della coppia è dovuta alle perdite dell'accoppiamento e all'inerzia del sistema (condizione di assenza di coppia resistente dovuta all'assenza di carico). I picchi che sono visibili rappresentano la coppia di accelerazione che è presente quando il sistema aumenta la velocità.

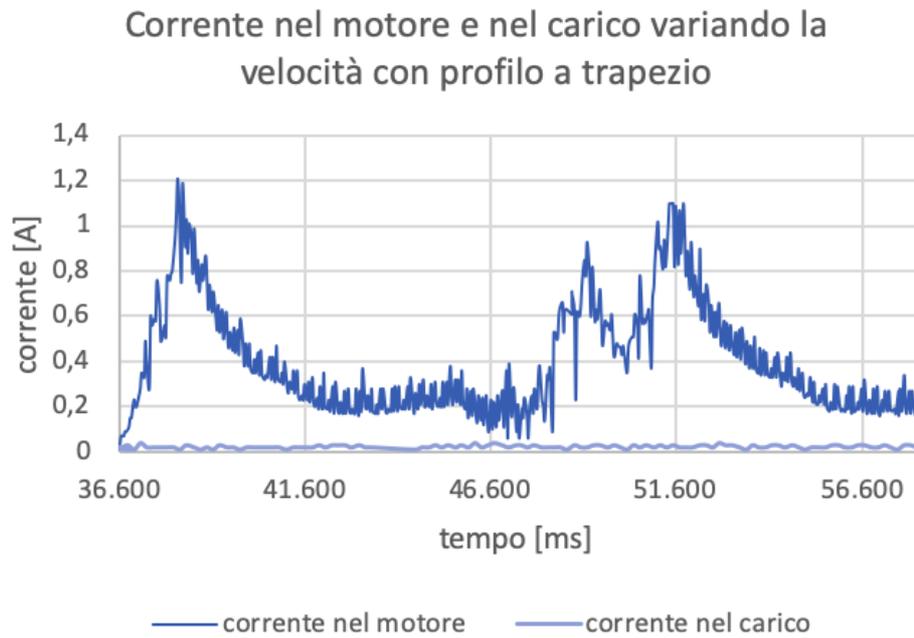


Figura 6.24: Confronto delle correnti lette nel motore e nel carico in un controllo PID di velocità con profilo a trapezio

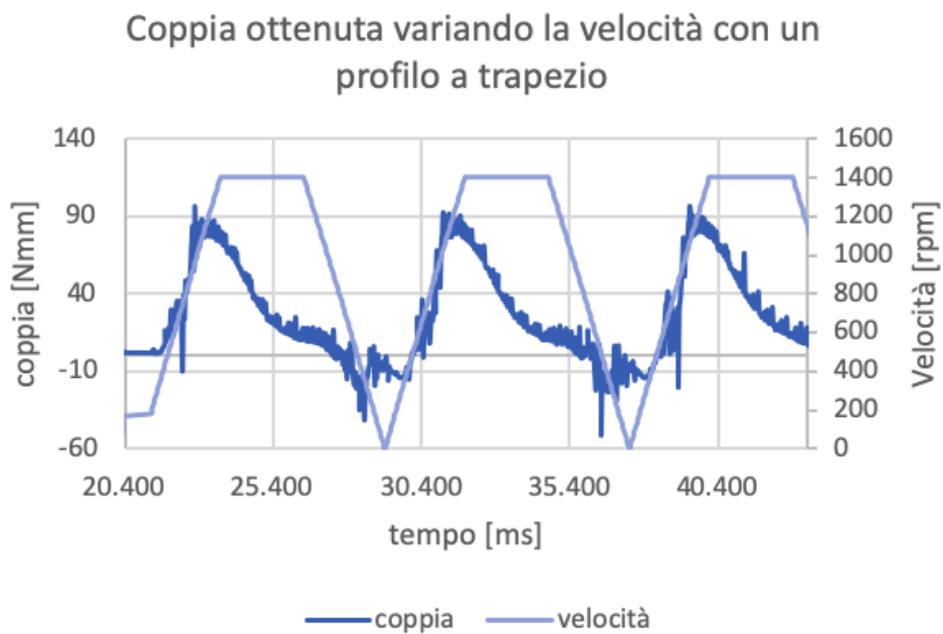


Figura 6.25: Coppia in un controllo PID di velocità con profilo a trapezio

Capitolo 7

Risultati sperimentali del controllo di coppia

In questo capitolo verranno riassunti i risultati sperimentali riguardanti il controllo di coppia. Verrà affrontato sia il caso di controllo ad anello aperto (open-loop), sia il controllo PID ad anello chiuso (closed-loop). Infine saranno implementate e testate alcune diverse tipologie di profili di coppia: profilo a step, profilo sinusoidale e profilo a trapezio.

7.1 Grafici del controllo di coppia ad anello aperto

In fig. 7.1 è stato testato un controllo ad anello aperto tramite la variazione del PWM in ingresso al motore. Il carico è stato impostato al massimo (circuito completamente chiuso). In questo modo è stato possibile visualizzare la coppia massima raggiunta a diversi livelli di tensione:

- tensione 10 V : 55 Nmm
- tensione 15 V: 61 Nmm
- tensione 17 V: 61 Nmm

La coppia nei casi di 15 e 17 V è uguale ma ciò che li rende diversi è la maggior oscillazione nel caso a 15 V. A 10 V la coppia è più bassa siccome la coppia dipende sia dalla tensione che dalla velocità del sistema.

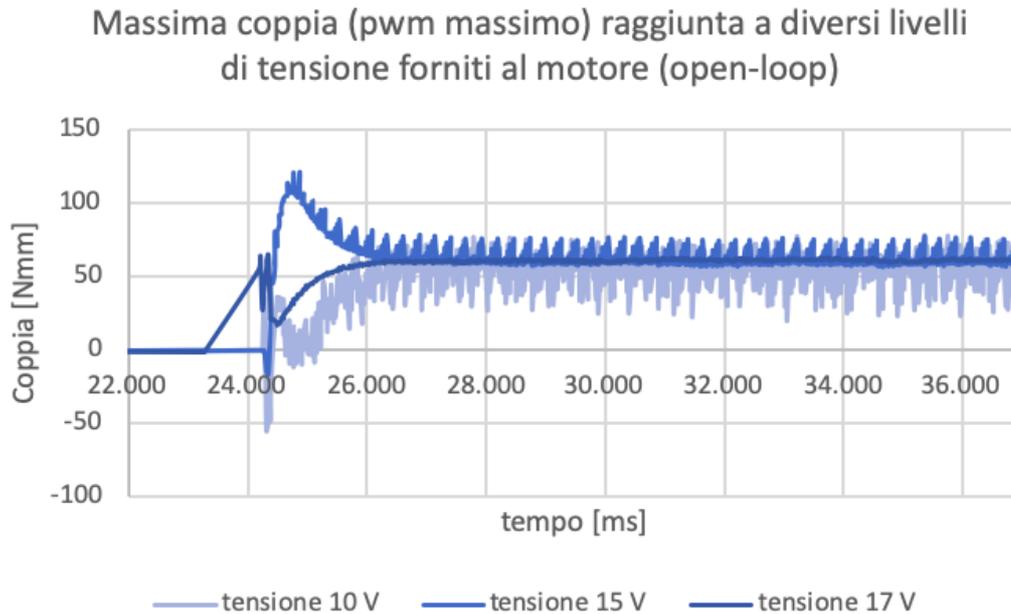


Figura 7.1: Massima coppia in funzione della tensione a cui viene alimentato il motore

In fig. 7.2 è stata rappresentata la variazione di velocità in un sistema alimentato ad una tensione fissa di 17 V, variando il PWM collegato al carico:

- PWM 0/255 : 1988 rpm
- PWM 50/255 : 1874 rpm
- PWM 150/255 : 1580 rpm
- PWM 255/255 : 1385 rpm

Si nota che la velocità a cui può ruotare il sistema aumenta con il diminuire del carico applicato. Nel grafico si evidenzia anche la presenza di piccole oscillazioni nelle caratteristiche intermedie: ciò è proprio dovuto al fatto che il PWM applicato al carico ha dei valori intermedi.

In fig. 7.3 è stato applicato un PWM nullo al transistor collegato al carico. In questo modo l'unica coppia che viene simulata è la coppia dovuta alle perdite dell'accoppiamento e all'inerzia del sistema. Infatti si può notare è che la coppia esistente in assenza del carico aumenta al crescere della tensione applicata. L'oscillazione maggiore nella caratteristica inferiore è dovuta a valori intermedi di PWM.

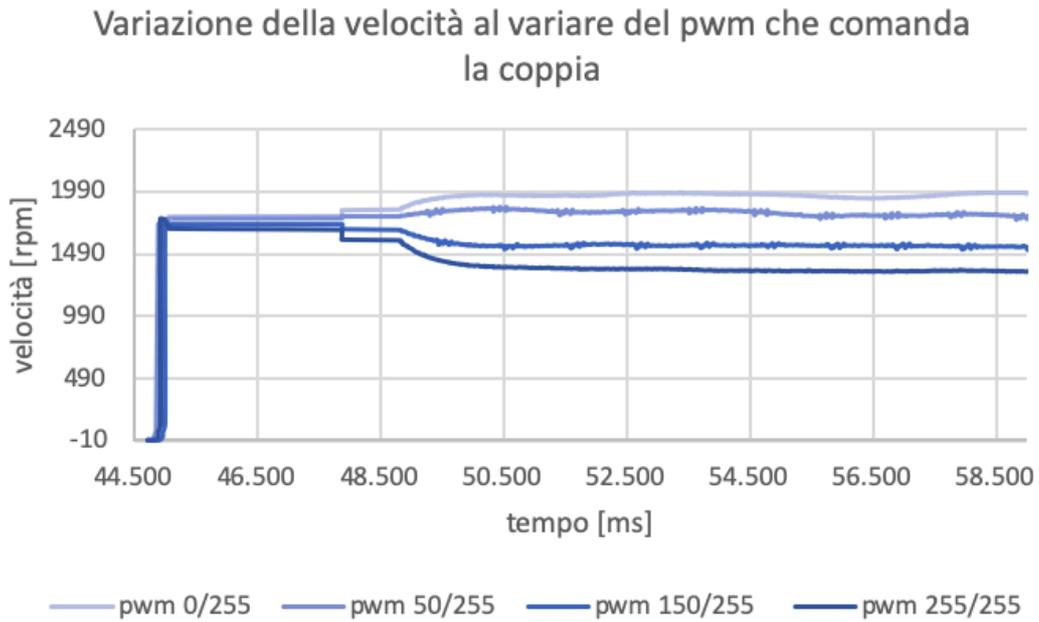


Figura 7.2: Variazione della velocità al variare del PWM che comanda la coppia

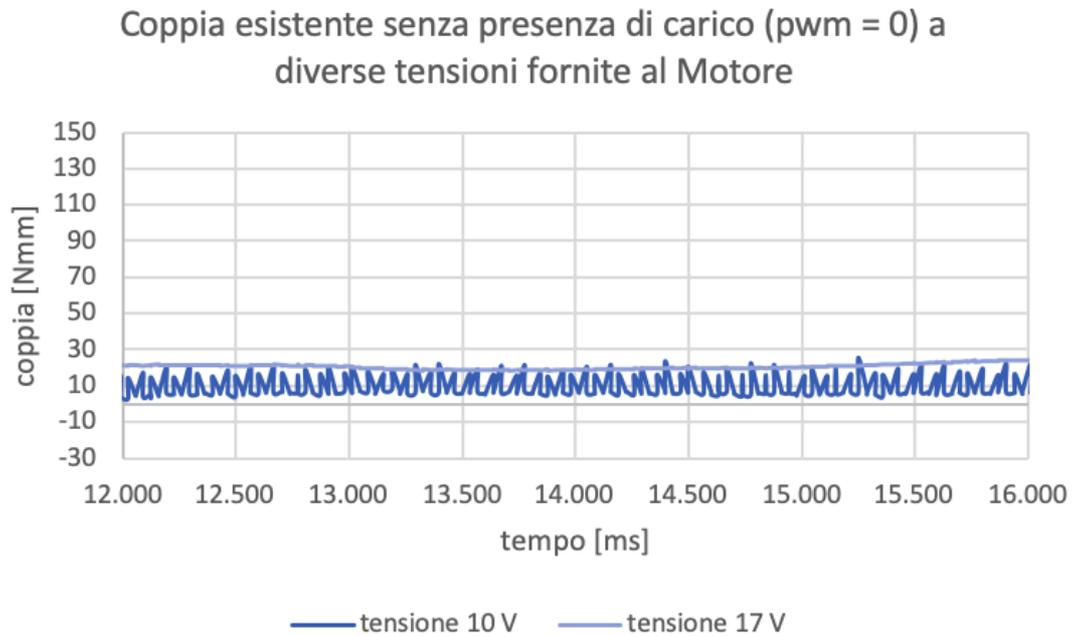


Figura 7.3: Coppia esistente (in assenza di carico) in funzione della tensione a cui viene alimentato il motore

7.2 Grafici del controllo PID di coppia (close-loop)

In questa sezione viene testato il controllo PID di coppia. I parametri usati dal PID sono:

- Pidtime = 0.01 ms
- $K_p = 1.5$
- $K_i = 10$
- $K_d = 0.4$

In fig. 7.4 è stata impostata la coppia di controllo pari a 30 Nmm e sono state variate le tensioni in ingresso al motore. Si possono notare oscillazioni crescenti al diminuire del PWM applicato al motore. Per questo motivo nelle sperimentazioni sarà spesso usata una tensione di 17 V al fine di avere andamenti più costanti.

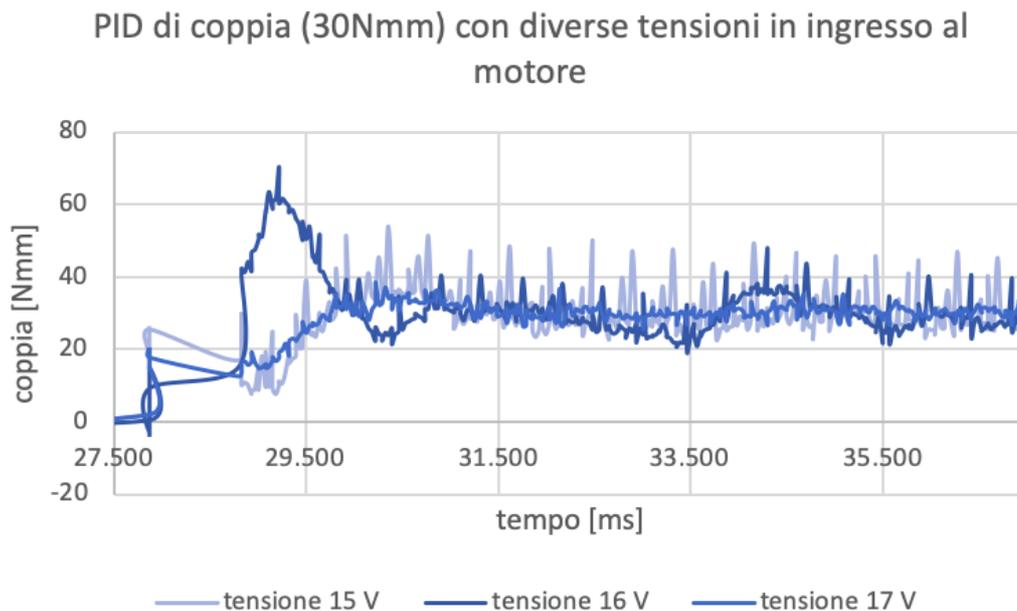


Figura 7.4: Controllo PID di coppia di 30 Nmm in funzione di diverse tensioni di alimentazione

In fig. 7.5 infatti viene alimentato il sistema con 17 V. La coppia impostata viene fatta variare tra 10 e 40 Nmm e si può notare che l'andamento della coppia controllata dal PID è molto simile a regime. Si possono avere diversi tipi di transitori: per esempio con la coppia maggiore si ha anche un overshoot elevato.

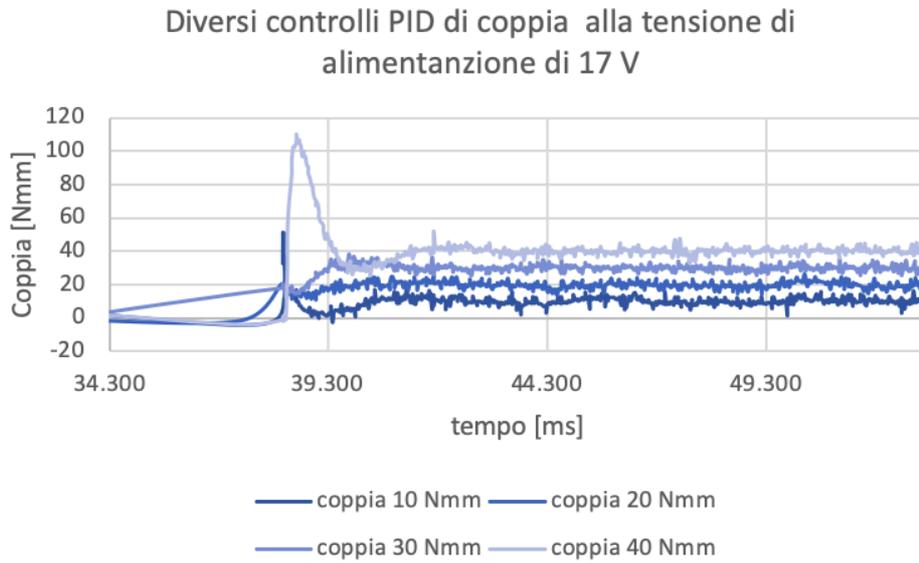


Figura 7.5: Diversi controlli PID di coppia

Anche la velocità risente dell'influenza della coppia. Come si può notare in fig. 7.6 la velocità è inversamente proporzionale alla coppia impostata.

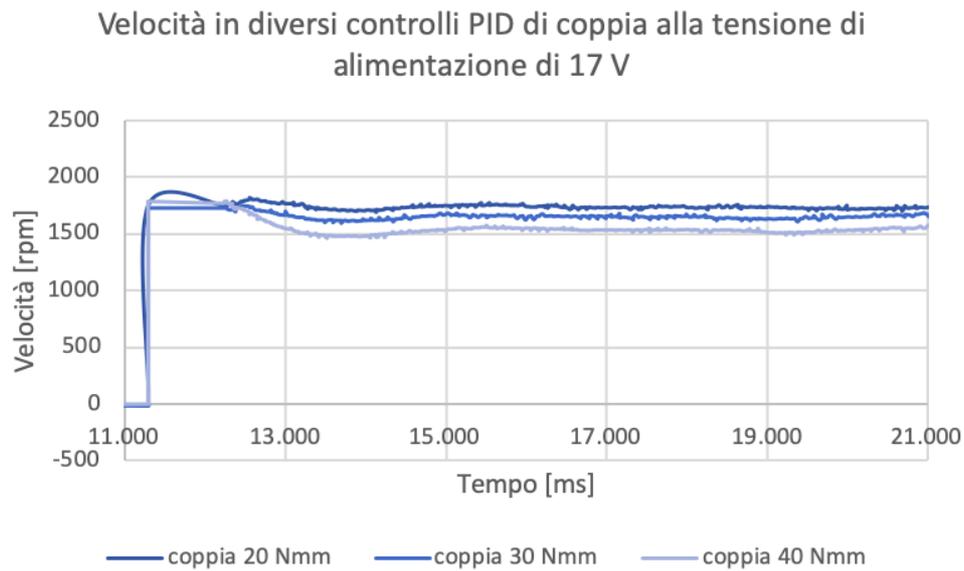


Figura 7.6: Velocità in diversi controlli PID di coppia alla tensione di alimentazione di 17 V

7.2.1 Grafici del controllo PID di coppia variabile (close-loop) con tensione di alimentazione fissa

In questa sezione è stato approfondito il confronto delle tensioni e delle correnti all'interno del sistema sottoposto a dei controlli PID di coppia, mantenendo una tensione di alimentazione fissa pari a 17 V e variando la coppia da controllare.

In fig. 7.7 sono confrontate le tensioni lette nel motore variando la coppia. Siccome la tensione nel motore non varia, si nota come sia indipendente dalla coppia. Invece le correnti nel motore, come si vede in fig. 7.8, dipendono dalla coppia con una proporzionalità diretta.



Figura 7.7: Tensioni lette nel Motore variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)

Le tensioni lette dal carico, in fig. 7.9, sono molto più oscillanti a causa della variazione di PWM comandata dal PID ma quasi coincidono al variare della coppia. Anche le correnti lette nel carico, in fig. 7.10, in questo caso hanno un andamento più oscillante per lo stesso motivo, ma la differenza rispetto alla tensione è un aumento della corrente al crescere della coppia.

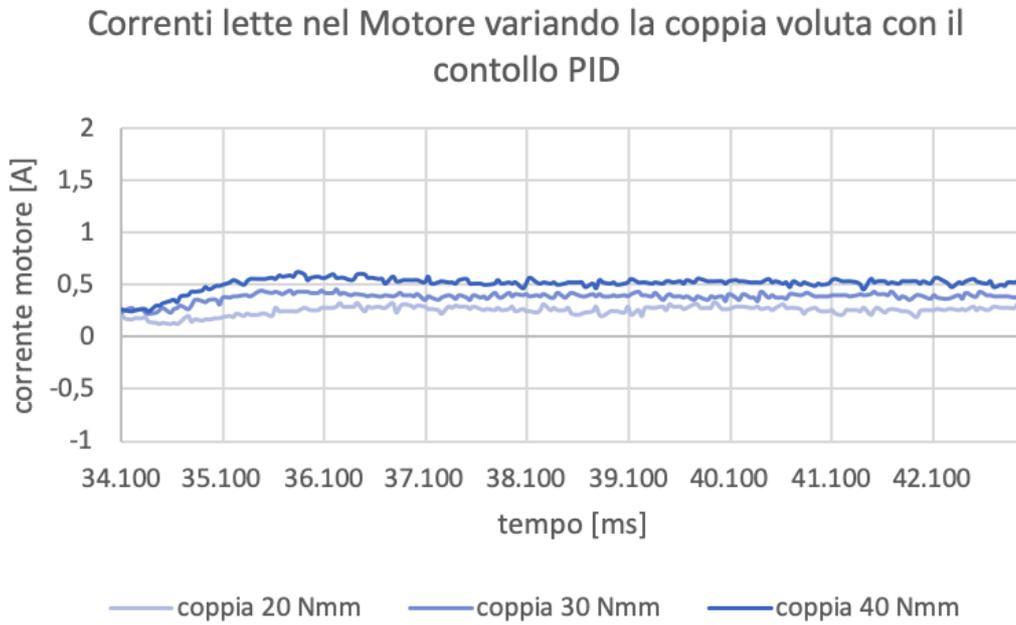


Figura 7.8: Correnti lette nel Motore variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)

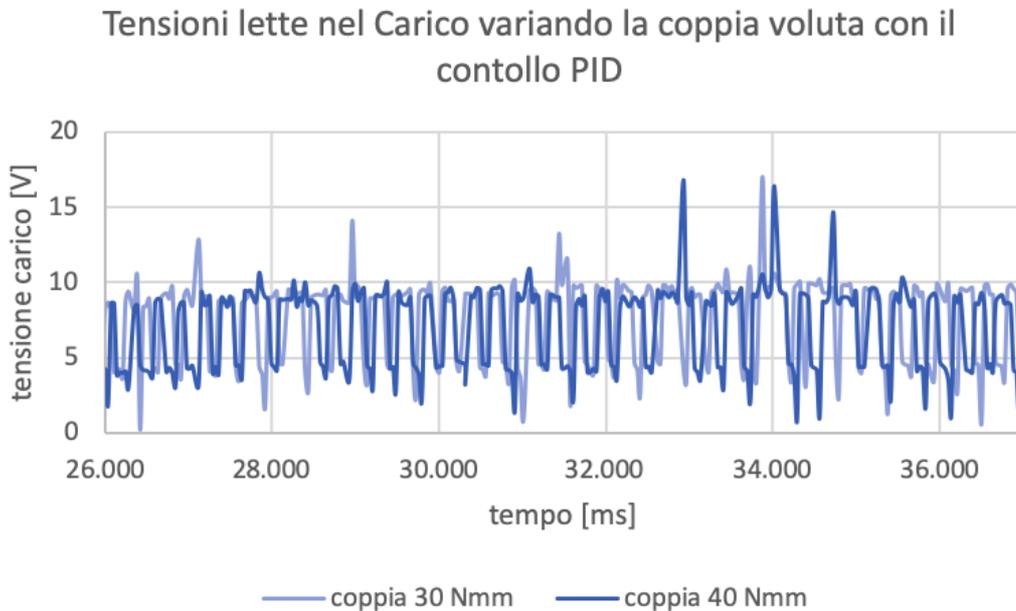


Figura 7.9: Tensioni lette nel Carico variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)

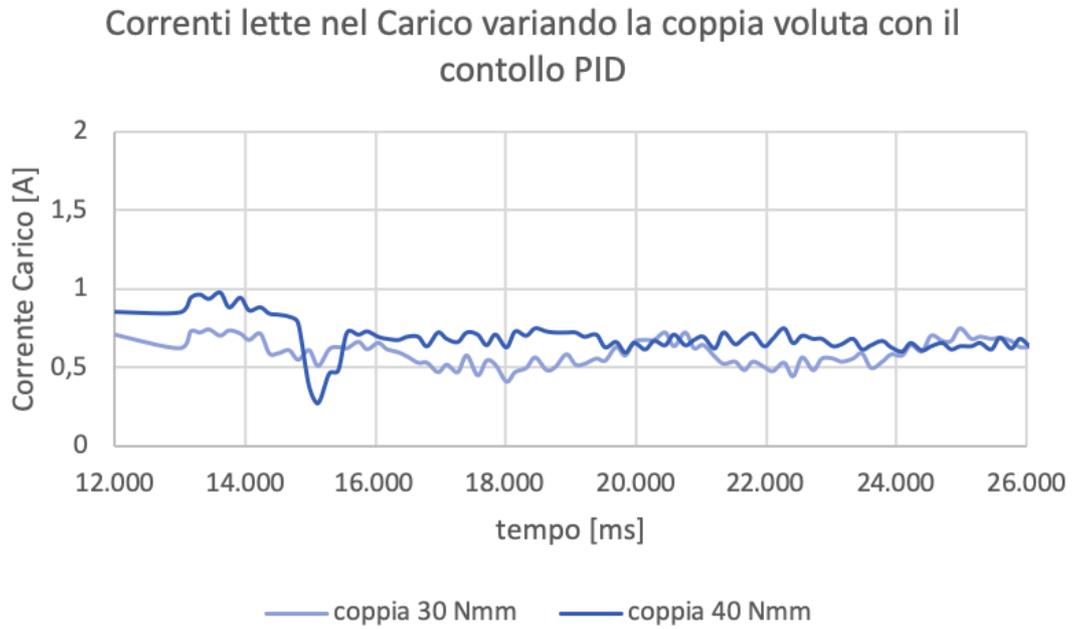


Figura 7.10: Correnti lette nel Carico variando la coppia voluta con il controllo PID (con tensione di alimentazione fissa di 17 V)

7.2.2 Grafici del controllo PID di coppia fissa (close-loop) con tensione di alimentazione variabile

In questo caso è stato effettuato il confronto tra tensioni e correnti imponendo al sistema un coppia fissa da controllare pari a 30 Nmm e variando la tensione in ingresso al motore.

In fig. 7.11 si nota che la tensione letta nel motore ha delle incongruenze con il valore dell'alimentazione con cui si sceglie di comandare il motore. Nel caso in cui si scelgano 10 V, si legge effettivamente la stessa tensione nel motore mentre nei casi in cui la tensione scelta è maggiore si hanno delle delle perdite nel circuito e ciò implica che la tensione percepita nel motore sia minore.

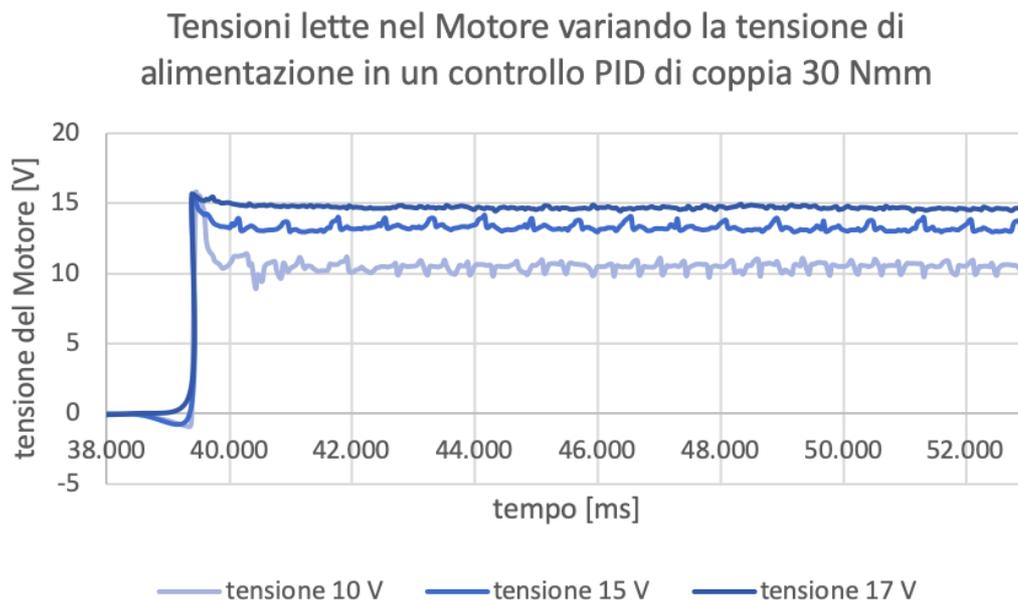


Figura 7.11: Tensioni lette nel Motore variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm

Le correnti lette nel motore, in fig. 7.12, dopo dei transitori simili vanno a coincidere sullo stesso valore di regime. Infatti la corrente letta nel motore è imposta dal carico.

Le tensioni lette nel carico, in fig. 7.13 hanno un andamento oscillatorio dovuto esclusivamente al controllo del carico. La tensione nel carico deve avere dei massimi più alti al crescere della tensione a cui è alimentato il motore.

Le correnti lette nel carico, in fig. 7.14 sono quasi costanti rispetto alla variazione di tensione all'ingresso dl motore.

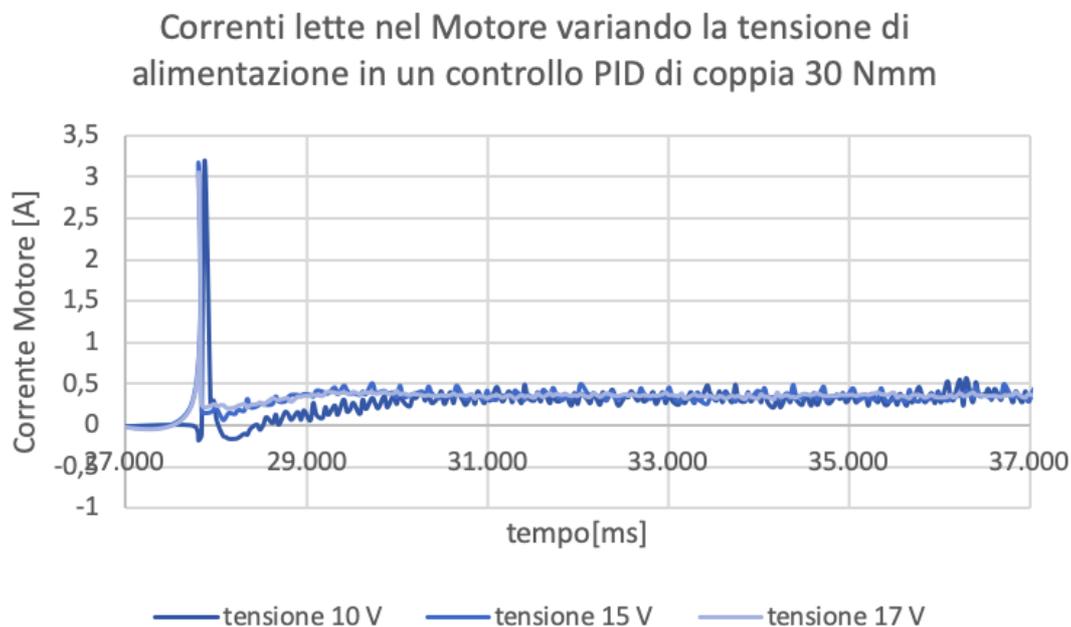


Figura 7.12: Correnti lette nel Motore variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm

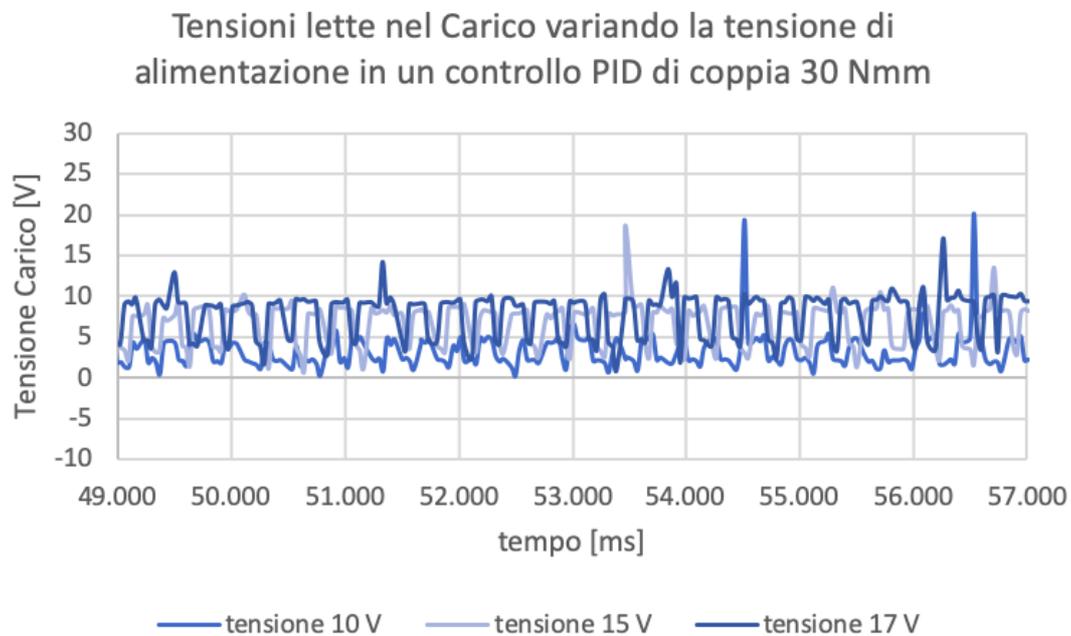


Figura 7.13: Tensioni lette nel Carico variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm

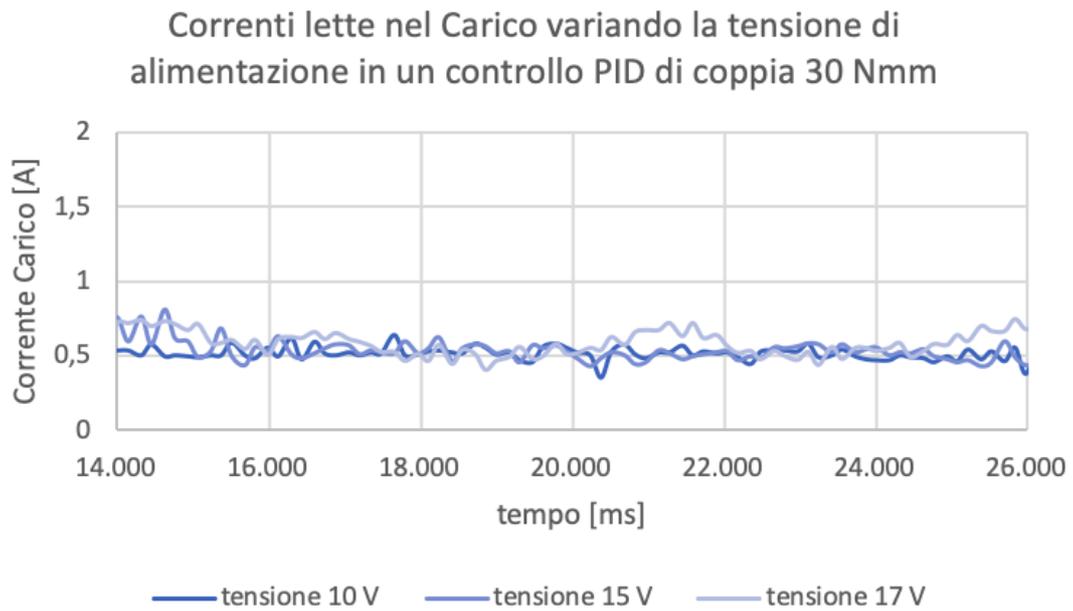


Figura 7.14: Correnti lette nel Carico variando la tensione di alimentazione in un controllo PID di coppia 30 Nmm

7.3 Controllo PID di coppia con profilo a step

I parametri usati nella creazione del profilo a step sono:

- Prima coppia impostata: 50 Nmm
- Seconda coppia impostata: 20 Nmm
- Tempo di avanzamento dello step: 3000 ms

In fig. 7.15, è riportato l'andamento del controllo PID di coppia con profilo a step. Il controllo mostra un certo overshoot nel passaggio tra la coppia minima e la coppia massima mentre, viceversa, nella transizione dalla coppia massima alla coppia minima l'assestamento è più preciso. Si evidenzia inoltre che la frequenza di variazione della coppia è troppo elevata e non permette al controllo di stabilizzarsi sui valori impostati di coppia: la dinamica della risposta del sistema richiede infatti un tempo di assestamento maggiore dei 3000 ms impostati.

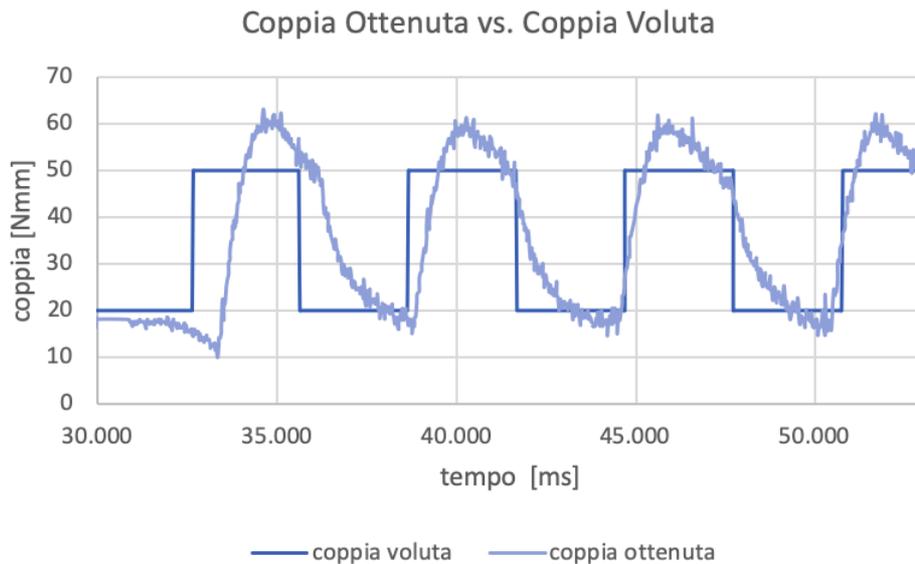


Figura 7.15: Controllo PID di coppia con profilo a step

In fig. 7.16, la tensione letta nel motore è confrontata con quella letta nel carico. La prima è più costante siccome è impostata da un PWM di 255/255, mentre la seconda è oscillante e dipende dal controllo PID.

In fig. 7.17, vengono confrontate le correnti lette nel motore e nel carico. Le due correnti hanno lo stesso andamento: seguono il profilo di coppia impostato sulla scheda di controllo sebbene ci siano ritardi che creano delle rampe di salita e discesa. La corrente del motore ha sempre un valore maggiore di quella del carico (ciò è visibile maggiormente alle coppie elevate).

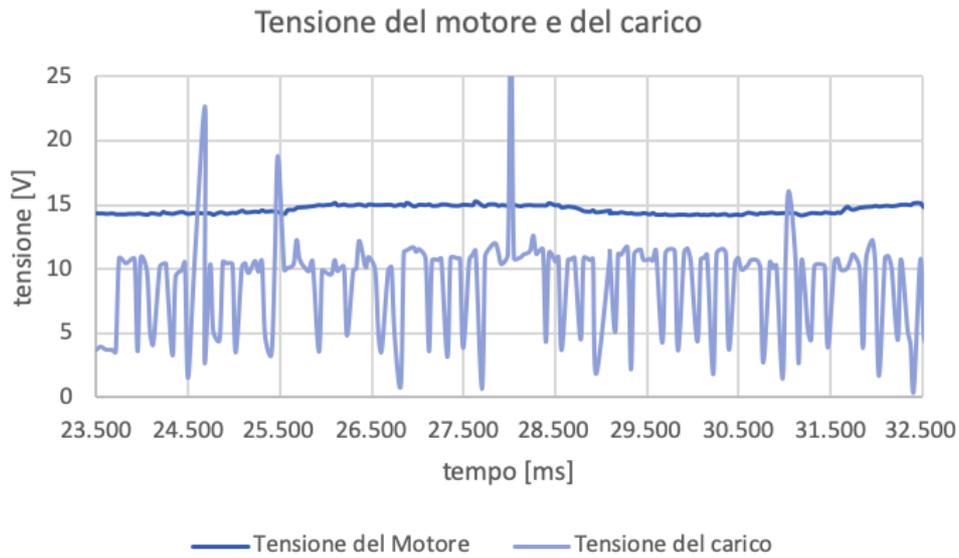


Figura 7.16: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a step

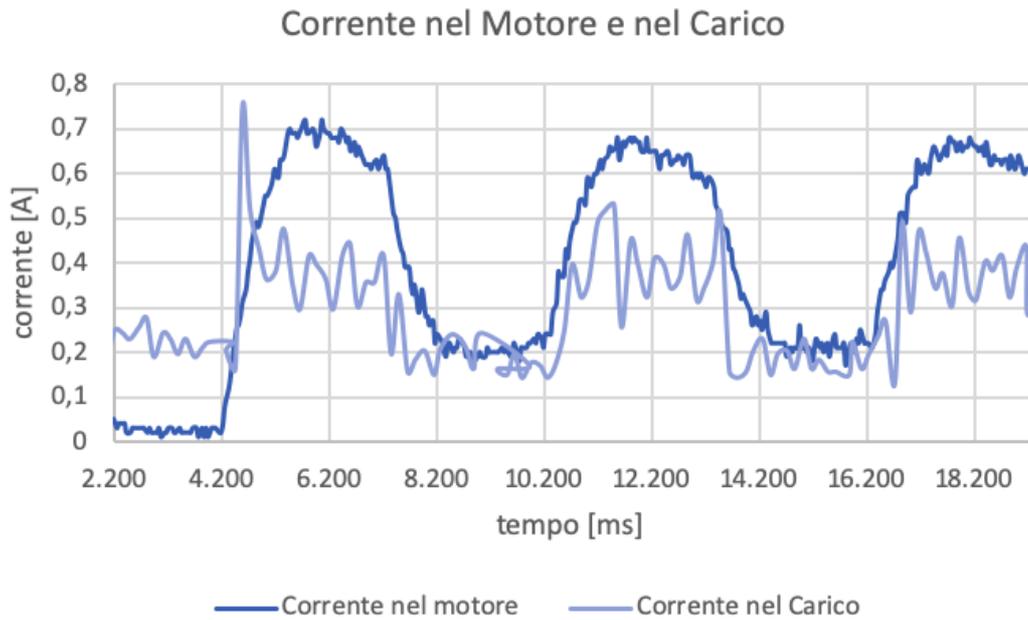


Figura 7.17: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a step

In fig. 7.18 è riportata la velocità che si raggiunge tramite il controllo PID di

coppia. In questo caso l'andamento della velocità è inversamente proporzionale alla coppia. Qualora la coppia aumenti, la velocità di rotazione del sistema deve diminuire.

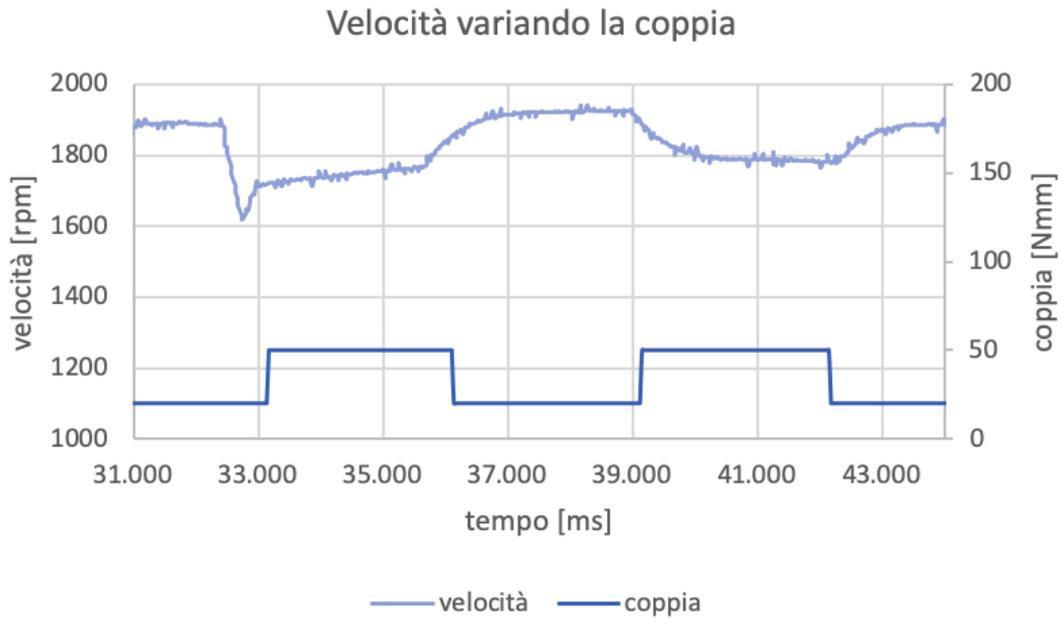


Figura 7.18: Velocità in un controllo PID di coppia con profilo a step

7.4 Controllo PID di coppia con profilo sinusoidale

I parametri usati per il profilo sinusoidale della coppia sono:

- Ampiezza del profilo sinusoidale di coppia: 40 Nmm
- Periodo del profilo sinusoidale: 4000 ms
- Tensione applicata: 17 V

In fig. 7.19 è presente l'andamento del controllo PID di coppia con profilo sinusoidale. Come nel caso del profilo a step, il primo ciclo è caratterizzato da un overshoot causato dall'errore accumulato nella fase iniziale, mentre i cicli successivi hanno un'andamento coerente, con la coppia che segue bene il riferimento impostato seppur con la presenza di piccoli ritardi nella risposta.

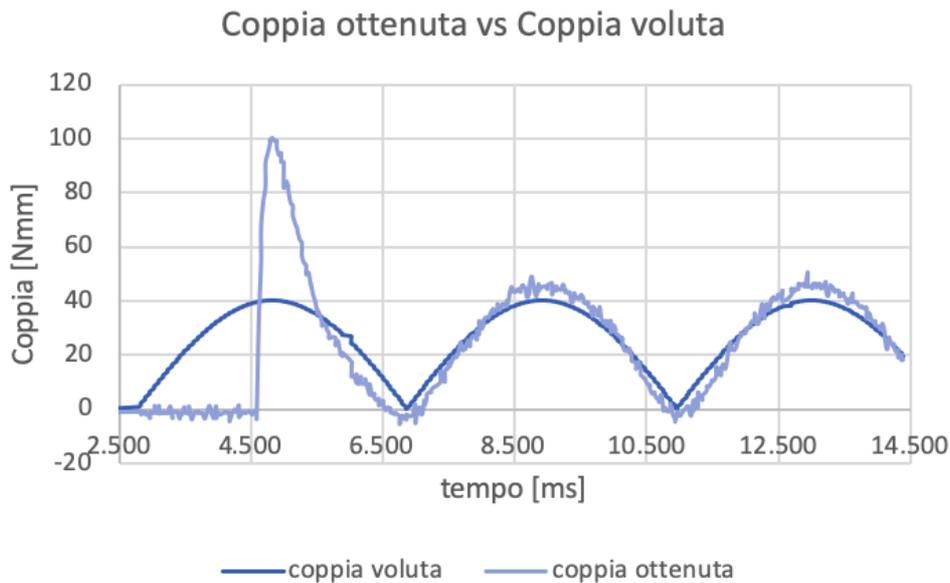


Figura 7.19: Controllo PID di coppia con profilo sinusoidale

In fig. 7.20 vengono mostrati gli andamenti delle tensioni nel motore e nel carico. Il motore ha una tensione sostanzialmente costante salvo leggere cadute di tensione dovute ai picchi di corrente che si hanno in corrispondenza dei picchi di coppia. La tensione letta nel carico è minore rispetto a quella letta nel motore e inoltre è più oscillante. Ciò è dovuto alla continua variazione di PWM applicato al carico lungo tutto il profilo.

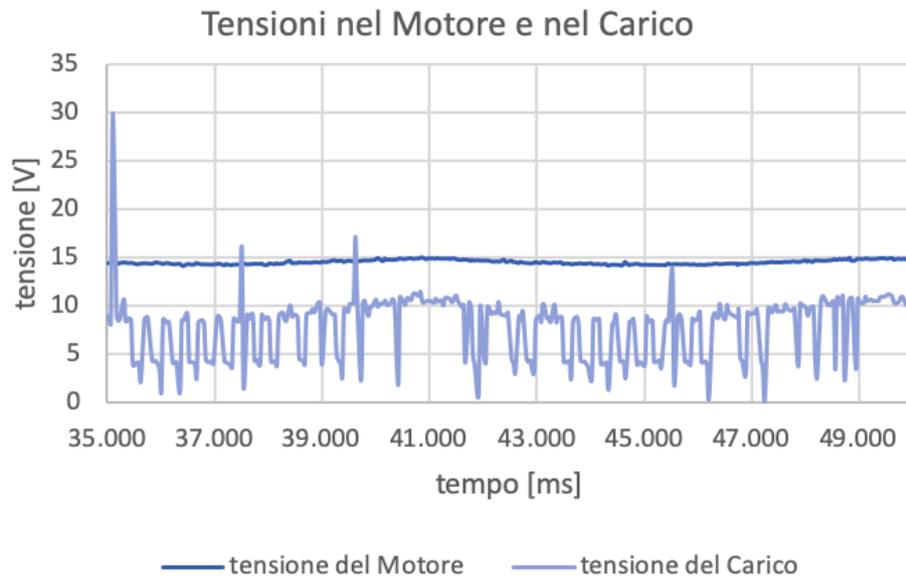


Figura 7.20: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo sinusoidale

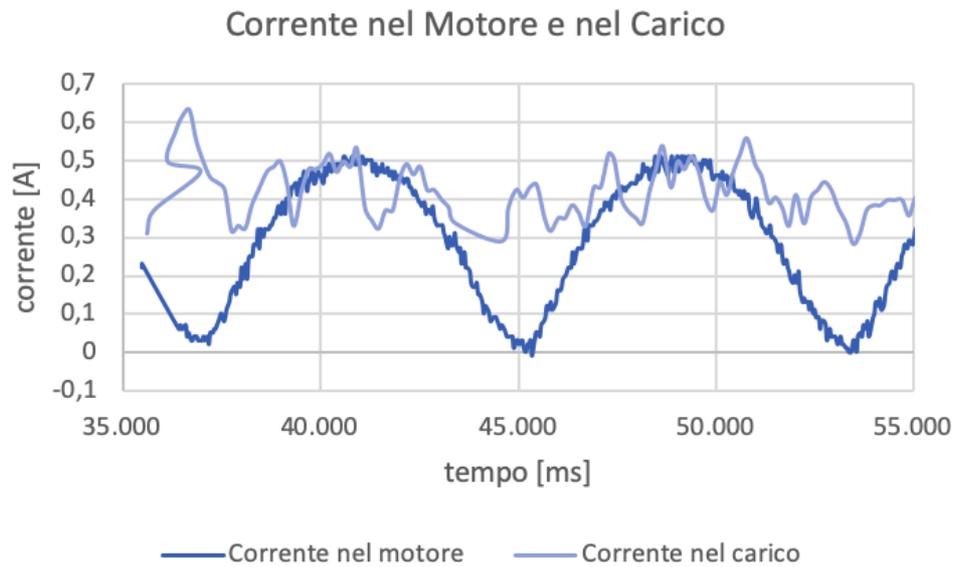


Figura 7.21: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo sinusoidale

Le correnti in fig. 7.21 hanno due andamenti diversi. Probabilmente in questo

caso il sensore di corrente collegato al carico non ha assicurato una lettura sensibile. La corrente letta nel motore invece ha un andamento coerente con il profilo di coppia ricercato.

In fig. 7.22 è stato riportato l'andamento della velocità raggiunta dal sistema in funzione dal profilo di coppia controllato. La velocità ha una proporzionalità inversa rispetto la coppia. Quando la coppia aumenta, la velocità deve diminuire.

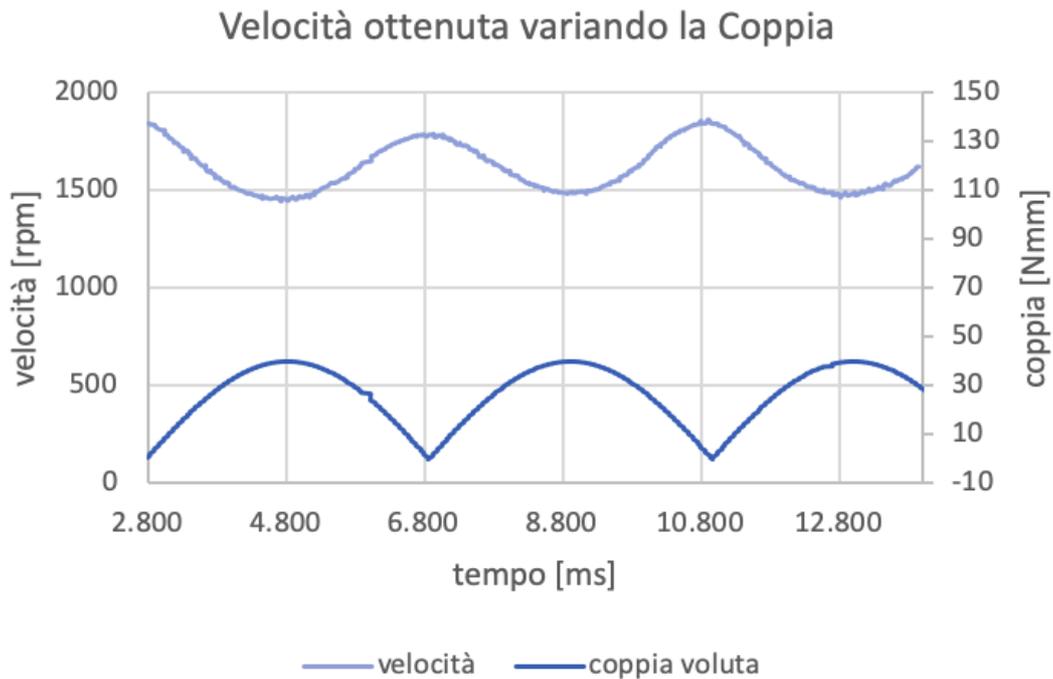


Figura 7.22: Velocità in un controllo PID di coppia con profilo sinusoidale

7.5 Controllo PID di coppia con profilo a trapezio

I parametri usati per simulare il profilo di velocità a trapezio sono:

- Coppia massima: 1400 rpm
- Tensione applicata: 10 V
- Tempo di salita: 100 ms
- Tempo a regime: 200 ms
- Tempo di discesa: 100 ms

In fig. 7.23 compare l'andamento del controllo PID di coppia con profilo a trapezio. Anche in questo caso il primo ciclo è un transitorio con un overshoot maggiore, mentre i cicli successivi hanno un'andamento più coerente con il profilo di coppia voluto, seppur siano presenti piccoli ritardi.

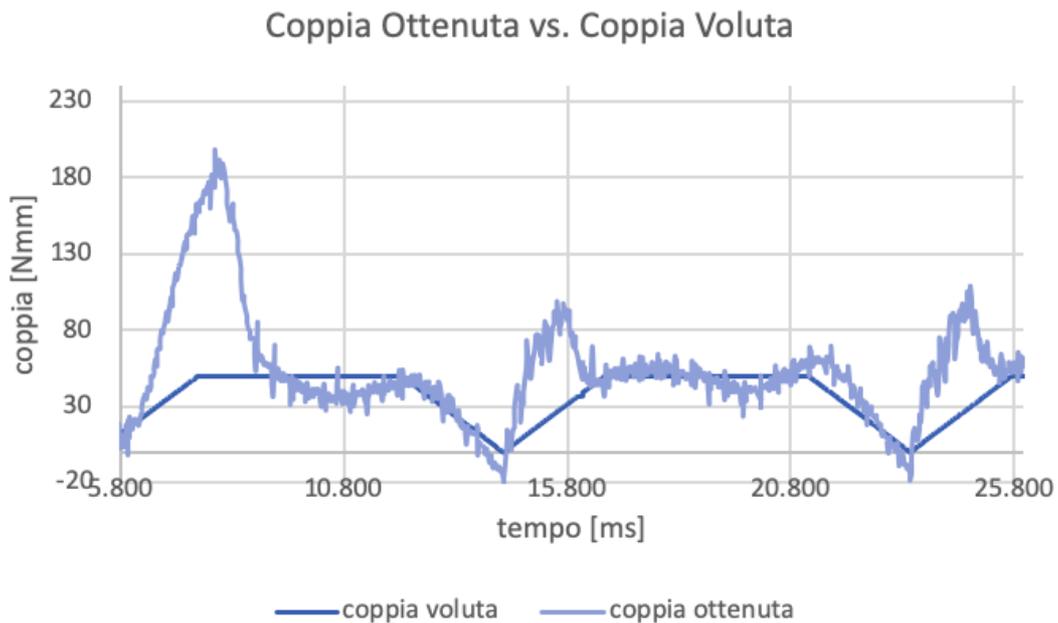


Figura 7.23: Controllo PID di coppia con profilo a trapezio

In fig. 7.24, è rappresentato il confronto tra le tensioni nel sistema. La tensione letta nel carico è minore rispetto a quella letta nel motore e inoltre è più oscillante. Ciò è dovuto alla continua variazione di PWM applicato al carico lungo tutto il profilo. Le oscillazioni aumentano quando la coppia cresce e viceversa.

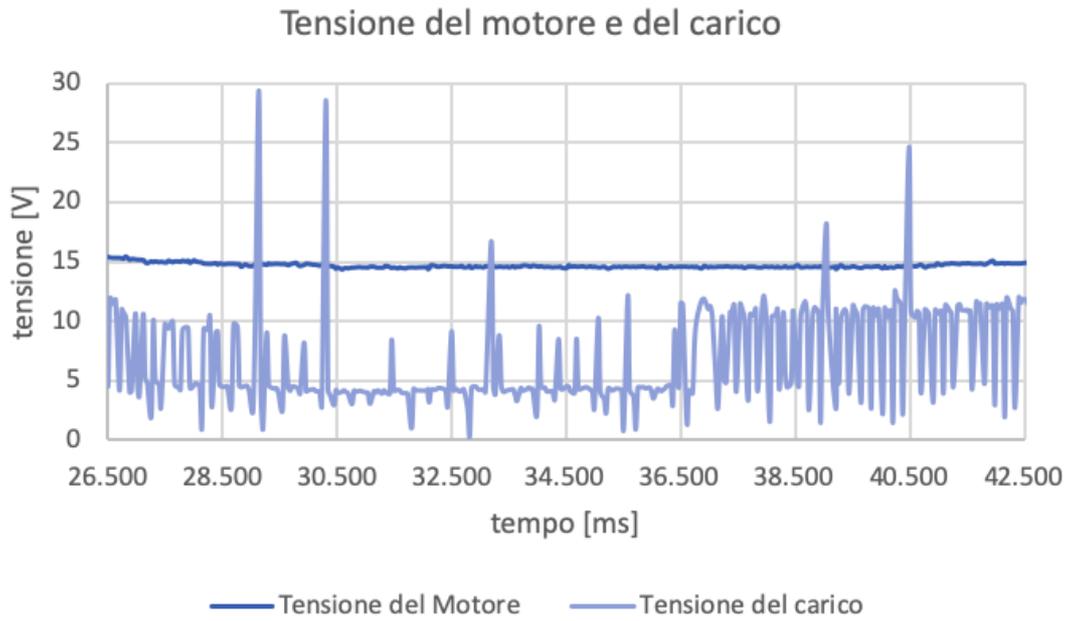


Figura 7.24: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a trapezio

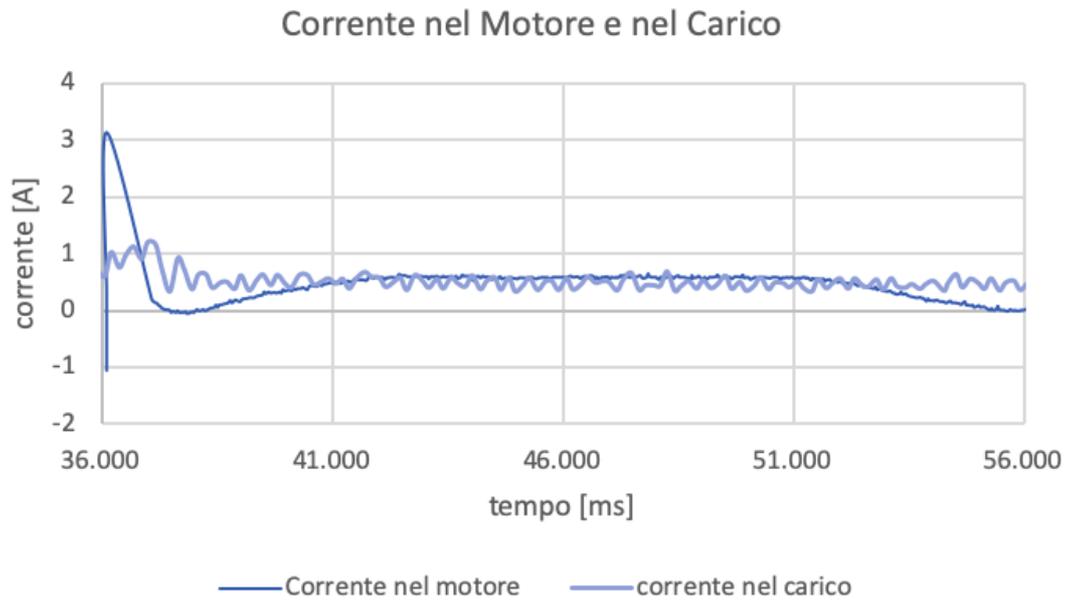


Figura 7.25: Confronto delle tensioni lette nel motore e nel carico in un controllo PID di coppia con profilo a trapezio

In fig. 7.25 è rappresentato il confronto tra le correnti del sistema. In questo caso la corrente nel motore è leggermente maggiore della corrente letta nel motore.

In fig. 7.26 è presente l'andamento della velocità di rotazione dell'accoppiamento in funzione del profilo di coppia controllato. La velocità ha una proporzionalità inversa rispetto la coppia. Infatti quando la coppia aumenta, la velocità deve diminuire.

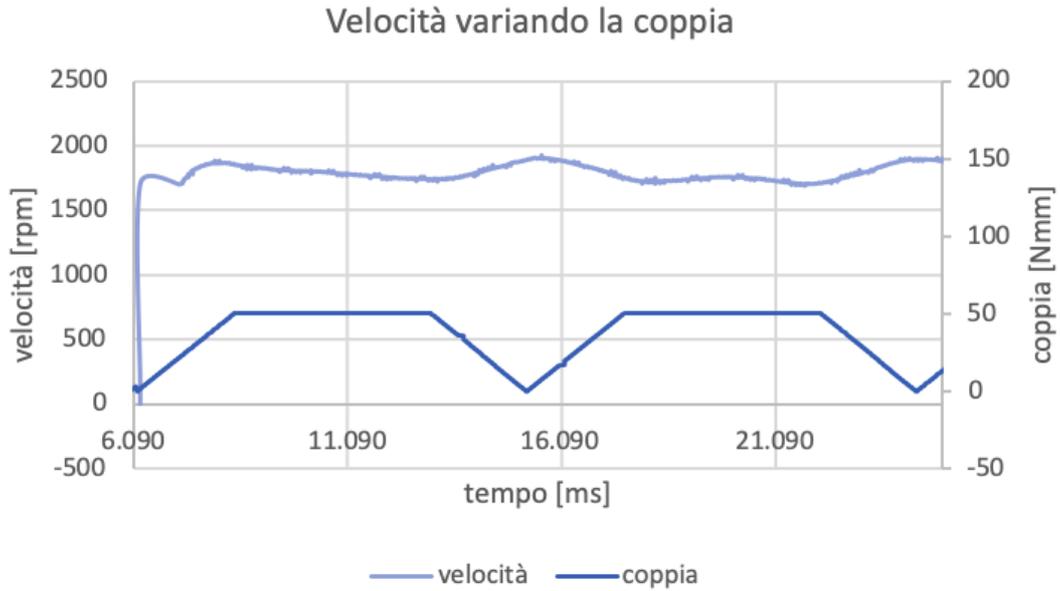


Figura 7.26: Velocità in un controllo PID di coppia con profilo a trapezio

Capitolo 8

Conclusioni

Il banco prova è stato sviluppato e testato a diverse velocità e a differenti valori di coppia. I risultati ottenuti confermano un buon controllo di velocità, mentre mostrano un controllo di coppia meno preciso.

Attualmente non esistono studi in letteratura, a conoscenza dell'autore, che mostrino come progettare un controllo di coppia con retroazione tramite il software Arduino.

Il controllo applicato in questa ricerca è stato formulato in maniera indiretta, sarebbe quindi utile, in un progetto futuro, l'adozione di un torsionmetro per un controllo diretto della coppia.

Poiché il fine ultimo di questa tesi è la progettazione un banco prova per l'accoppiamento tra alberi scanalati, si prevede di implementare il codice e lo studio scalandolo su una potenza maggiore. Ciò potrà essere effettuato tramite l'uso di un encoder al posto della dinamo tachimetrica. Incrementando quindi la precisione e la grandezza del banco si andrebbe di conseguenza a confermare o meno la bontà dei risultati. Approfondire questo studio, sottolineando l'importanza di una potenza maggiore disponibile, sarebbe interessante al fine di avere un visione più specifica e ampia del problema.

Bibliografia

- [1] Hughes, A., cur. *Electric Motors and Drives*. Elsevier, 2006.
- [2] L.Pugi, B. Allotta e, cur. *Meccatronica, azionamenti elettrici e oleodinamici*. Esculapio, 2016.
- [3] S. Fitzgerald M. Shiloh, M. Piccin, cur. *Meccatronica, azionamenti elettrici e oleodinamici*. Arduino AG, 2017.
- [4] V. Viktorov, F. Colombo e, cur. *Automazione dei sistemi meccanici. Corso di base*. CLUT, 2013.
- [5] Ilardo, R. *Breve guida all'uso dei transistori*. URL: <http://www.raffaeleilardo.it/trans.htm>.
- [6] Nee. *Guida Arduino 10: Le uscite PWM*. 2018. URL: <https://www.ne555.it/guida-arduino-uscite-pwm/>.
- [7] Beauregard, B. *PID Library*. 2017. URL: <https://playground.arduino.cc/Code/PIDLibrary/>.
- [8] Maffucci, Michele. «Appunti su Arduino: interrupts». In: (giu. 2012).
- [9] J. Heidenhain, dr. *Rotary Encoders*. 2017.
- [10] Yap, N. *digitalWriteFast*. 2015. URL: <https://github.com/NicksonYap/digitalWriteFast>.
- [11] Ravenna, Itis. *Encoder*. URL: https://itisravenna.edu.it/corso/labsitel/provelab/caratt_motcc/encoder.htm.
- [12] unimore. *Sensori e trasduttori*. URL: <http://www.automazione.ingre.unimore.it/pages/corsi/materialeDidattico/itsc0809/itsc09-sensori.pdf>.
- [13] Villa, V. *Dinamo tachimetrica ed encoder*. 1998. URL: <https://www.vincenzov.net/tutorial/motoridc/dinamo.htm>.
- [14] unife. *Azionamenti elettrici*. URL: <http://www.unife.it/ing/informazione/Azionamenti-elettrici/materiale-corso/lez5-10.pdf>.
- [15] Arduino. *Language Reference*. URL: <https://www.arduino.cc/reference/en/>.